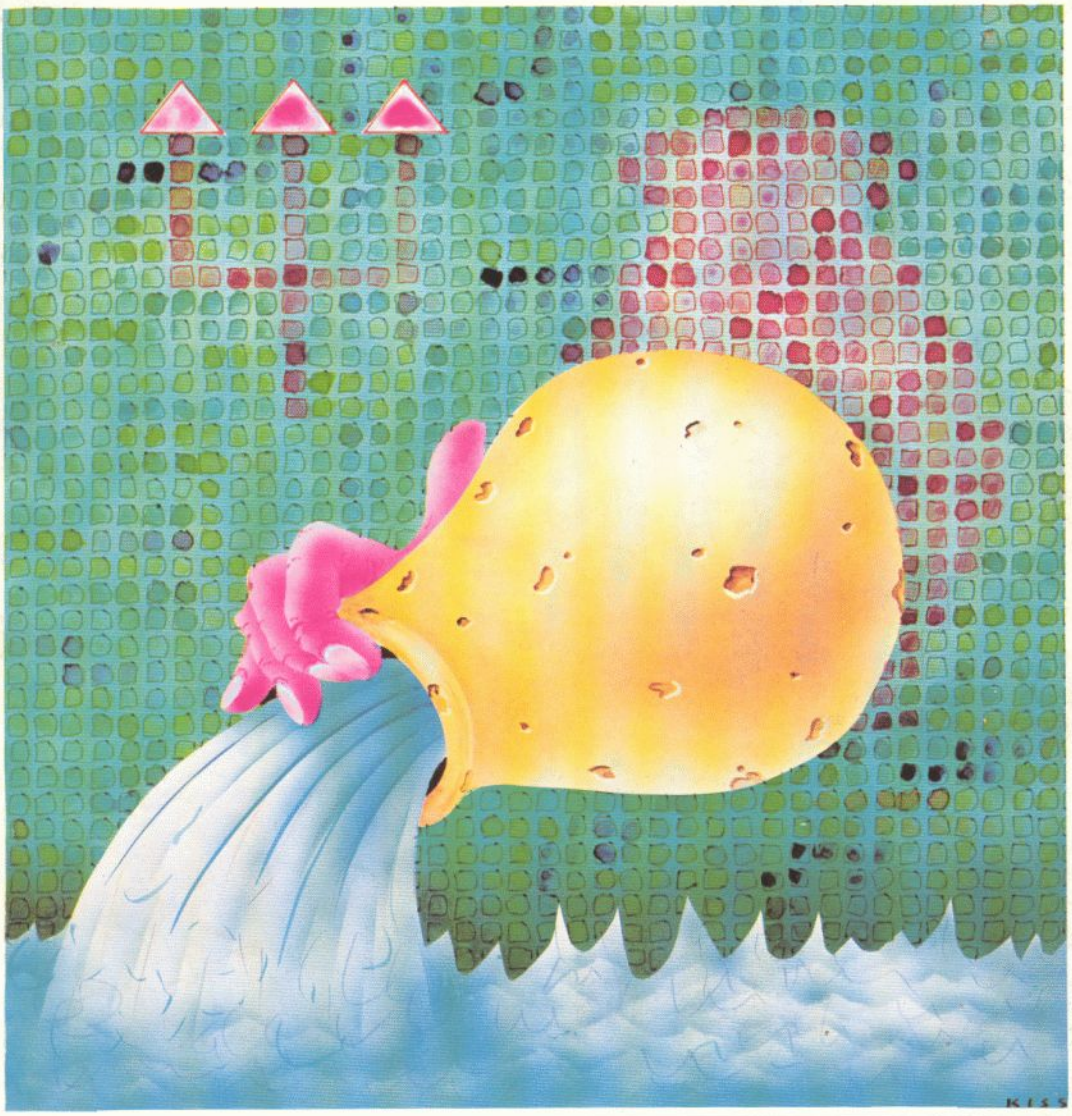
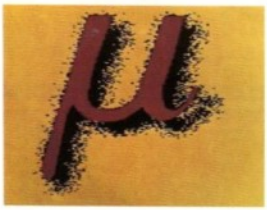
Ára: 30 Ft

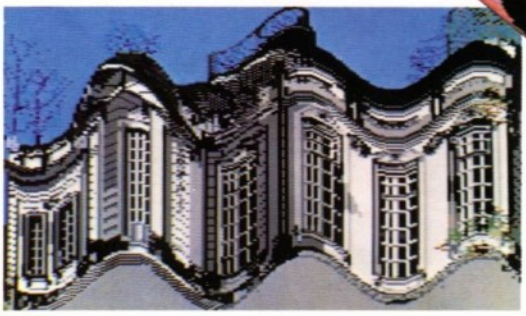
μ mikro számítógép magazin



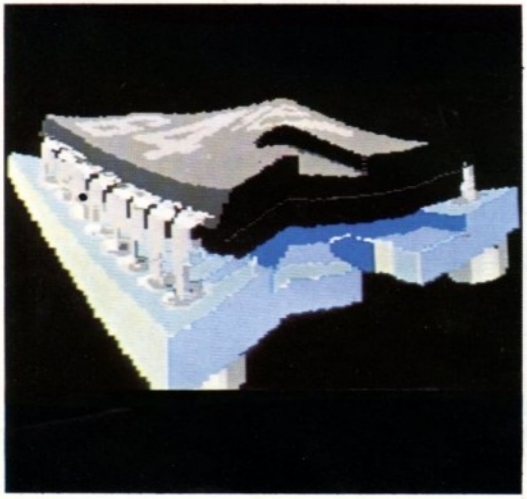
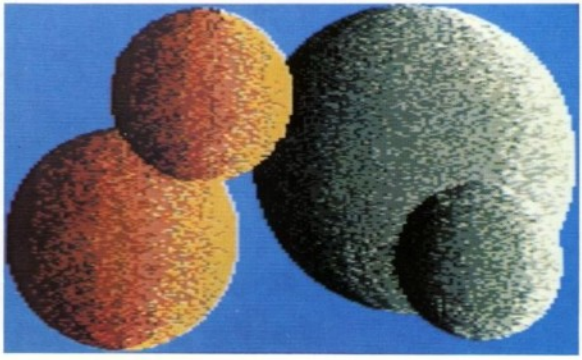
VÍZ(D)ÖNTÉS

1988/11





ECSET NÉLKÜL

mikro számítógép magazin

6. ÉVFOLYAM
1988/11. SZÁM

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG LAPJA

A szerkesztőbizottság vezetője:
Kovács Győző

A szerkesztőség munkatársai:
Bakos Tamás (programozástechnika)
Broczkó Péter (hírek)
Ferencz Mária (tervezőszerkesztő)
Kovács Győző (levelezés)
Petróczy Judit (könyvek)
Pinke György (NJSZT, alkalmazások)
Simonyi Endre
Szebenszki Sándor
Szulyovszky Csaba
Tamásné Lakó Erika
Terebessy Ákosné
Vizesi Mária

Címképünk:
Kiss Ilona munkája

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Király G. István
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlap-előfizetési és Lapellátási Irodáján
(1900 Budapest XIII., Lehel u. 10/A)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta.
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, Pf. 149.
és a Magyar Média
1932 Budapest, Pf. 279.
88—1552

Szikra Lapnyomda
Budapest (88—1552)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

TARTALOM

	2	ECCE '89, Lausanne
	5	Feladatok — megoldások
	17	A nagyfelbontású képernyő forgatása bájtonként I.
	19	Mindent a vevőért?
	20	Fényújság FORTH-ban
	21	Víz — kvíz
	22	Egy gép száz bajt csinál?
	28	Rendszerfejlesztési eszközök
	32	Mi van a cellában?
	34	Merre tart a világ?
	37	Semmi sem lehetetlen!
	39	µ INFORM
	40	Olvastunk . . .
	44	Egy sarokkal olcsóbb!
	45	Programtermék
	46	Adok—veszek—cserélek
ISKOLASZÁMÍTÓGÉP	**3**	
	3	Mérés és irányítás számítógéppel
CSIPEGETŐ	**7**	
	7	Miért ezt, s miért nem mást?!
	8	Szinte egy lélegzetre . . .
	9	Tanulás — siker — tábortűz
	10	RENEW C16-ra, TOP-lista
PROGRAMOZÁSTECHNIKA	**12**	
	12	BASIC és gépi kód
	13	Néhány nyelvről — néhány szóban
	15	A programozás alapkérdései III.
ENTERPRISE	**24**	
	24	Számábrázolás és -tárolás
	25	EXOS 2.1
	26	Téglatest a képernyőn
	27	Megkérdeztük az Enterprise-ról
µKLUB	**38**	
	38	Adom a magyarázatot
	39	Egy- és kétsorosok
SAKK	**42**	
	42	A sakkprogramozókat segítő statisztikai adatok
AZ OLVASÓ ÍRJA	**43**	
HÍREK—ÉRDEKESSÉGEK—KÖNYVEK	**46**	
PONTVADÁSZAT	**48**	

ECCE '89, Lausanne

European Conference on Computer in Education (A számítógép az oktatásban — európai konferencia)

„A számítógép feltalálása az emberiség történetének valódi forradalmát jelentette, főleg azért, mert amióta ember egyáltalán létezik, még nem volt képes ennyi szellemességet, ilyen rövid idő alatt papírra nyomtatni.

(Ismeretlen angol humorista írása szabad fordításban)

Az IFIP TC 3 (az IFIP az információfeldolgozás nemzetközi szervezete: International Federation of Information Processing) fennállásának 25. évfordulóját az első európai számítógépes oktatási konferencia megrendezésével ünnepelte.

A megnyitó előadást *W. Brauer*, a TC 3 elnöke tartotta, aki emlékeztetett arra, hogy a TC 3, a számítógépes oktatással foglalkozó IFIP műszaki bizottság a nemzetközi szervezet egyik legrégibb szakbizottsága.

Egy rövid írás keretében nyilvánvalóan nem lehet a konferenciáról részletesen beszámolni. Ezért elsősorban azokat az előadásokat válogattam, amelyek az új számítógépes tanítási (tanulási) technológiákról, a nemzeti számítógépes programokról, a courseware-fejlesztésről, a számítógépes tanulás szociális, társadalmi hatásáról és persze a távtanulásról szóltak, másrészt mintegy idézetgyűjteményt állítottam össze a beszélgetéseimből.

Számítógéppel támogatott tanítás (CAI) vagy tanulás (CAL)?

Látszólag terminológiai problémáról van szó, mégis nagyon sok beszélgetésnek volt a témája, hogy minek is nevezzük azt a technológiát, amiről beszélünk és amit alkalmazunk.

A CAI a tanításhoz használt számítógépes demonstrációs anyag, voltaképpen igen korszerű szemléltető eszköz, amit a tanár használ a tanteremben, segítségként a tananyag megértetéséhez.

A számítógépes tanulás (CAL) céljára készített tananyagot a tanuló (felnőtt vagy gyerek) általában tanári közreműködés nélkül, esetleg más tananyaghordozókkal (könyv, film, video stb.) együtt, önállóan használja.

Tehát már ebből a rövid összevetésből is nyilvánvalóan kitűnik, hogy ugyanannak a témának a CAI és a CAL változata pedagógiailag egész biztosan különböző.

A számítógép az iskolában

Az elemi és a középiskolák számítógépesítését általában nemzeti programok keretében oldották meg. Azokban az országokban, ahol a programot hosszú távra és átgondoltan tervezték, ott a számítógép az oktatás és a tanulás hatékonyságát jelentékenyen megnövelte. Azokban az országokban azonban, ahol az „olcsón vásárolni és sokat" szemlélet volt az uralkodó, ott ma az a gond, hogy mire is használják a semmire sem alkalmas „játék" számítógépeket. Ezekre a gépekre courseware-t nem érdemes fejleszteni, kommunikációra alkalmatlanok, még az osztálytermi demonstráció is majdnem lehetetlen megfelelő szoftver hiján.

Sajnos, ez a mennyiségi szemlélet — a sokat és olcsón — leginkább a szocialista országok iskolaszámítástechnikájára jellemző, aminek az eredményeképpen nálunk még sokáig várni kell arra, hogy iskolai és iskolák közötti információs rendszereket, lokális, regionális vagy nemzeti iskolai hálózatokat szervezzünk.

A courseware-piac

Több előadás foglalkozott a számítógéppel támogatott tanítás és tanulás gazdasági jelentőségével.

Az Egyesült Államokban például a CAI és CAL termékek már ma is igen előkelő helyet foglalnak el piaci sikerüket nézve a számítógépes produktumok között. Gazdasági szakemberek becslése szerint a következő évtizedben a CAI és főleg a CAL piaci forgalomban az élre kerül, aminek persze az a feltétele, hogy az oktatási rendszer is a modern technológia követelményeinek megfelelően változzon. Ez azt jelenti, hogy az iskolai oktatás helyett az otthoni tanulás legyen a tudás megszerzésének az első számú lehetősége.

Tanítsuk vagy ne tanítsuk a programozást?

A konferencia előadóit hallgatva, de még inkább beszélgetve a résztvevőkkel, úgy láttam, hogy a vita: tanuljanak-e a gyerekek programozni vagy sem, egyáltalán nem dőlt még el.

Jaques Heberstreit megnyitó előadásában kétségét fejezte ki, hogy hasznos-e ma a programozás tanítását erőltetni, hiszen ma még nincsenek igazán reális elképzeléseink arról, hogy tíz év múlva, amikor a mai gyerekek felnőttek lesznek, milyen számítógépes környezetben kell dolgozniuk. Lesz-e még akkor az a valami, amit ma programozásnak nevezünk?!

Persze az ellentábornak is van elegendő „muníciója": ők nemegyszer éppen *Seymour Papertre* (ő nem volt ott) hivatkoztak, mondván, a számítógépes munkánál semmi sem tanítja meg jobban a gyereket, hogyan kell gondolkodni, márpedig ma a számítógépen való munka többé-kevésbé egyenlő a programozással.

Heberstreit véleménye ezzel a felfogással nem egyezik, azt mondja, hogy a gyárakban, vállalatoknál, de még az iskolákban sem az ma a szakemberek dolga, hogy programozzanak, hanem hogy dolgozzanak, s használjanak mindenhez számítógépet. Már van is alkalmas technológia, amelyet úgy hívhatunk, hogy CAX, azaz Computer Assisted X, vagy magyarul számítógéppel támogatott „izé" (tevékenység), ahol az X lehet Learning (CAL) vagy Instruction (CAI), tehát tanulás vagy tanítás, avagy más, és lehet, hogy a tíz év múlva dolgozni kezdő szakembernek valóban nem lesz már szüksége arra, hogy programozzon, elég, ha a számára legalkalmasabb CAX-eszközt kiválasztja és azzal oldja meg a feladatát.

A végére hagytam a *távtanulást*, ami, úgy vélem, jelentőségének megfelelően, igen nagy helyet kapott mind a programban, mind a plenáris előadásokban.

T. Stonier (Nagy-Britannia) megfogalmazása szerint a technologizált világnak egyre több tanult emberre van szüksége. Az ember, mint tőke, a mai modern gazdaságnak fontosabb tényezője, mint bármikor korábban volt, és az emberi tőkét egyre kevésbé az ember fizikai, sokkal inkább a szellemi teljesítménye vagy áttételesen a képzettsége határozza meg, vagyis a tanulás jelentősége a jövőben egyre nő.

T. Stonier azt állította, hogy a fejlett társadalmak már ma is nemzeti jövedelmük jelentős hányadát, a jövő században pedig már a legnagyobb részét az „egyes számú nemzeti beruházásra", a tanítási és főleg a tanulási rendszerük fenntartására és továbbfejlesztésére fogják fordítani. Következményként pedig kialakul egy új társadalomkép, a tanult emberek társadalma, amelyben az elsőrendű érték a tudás.

Ebben a társadalomban — mondja *T. Stonier* — a tanulás, az önképzés élethosszig tart, és persze mindenki örömmel tanul, hiszen olyan ismereteket szerez, amelyekkel nemcsak szellemileg gyarapodik, de a pályáján is előbbre jut. Igaz, azt is hozzátette, hogy ma ez még csak álom, de a megvalósítás már közelebb van, mint gondolnánk, hiszen a modern képzési technológia eszközei már adottak, és jó néhány országban már nyílt oktatási rendszerek (például Open University) formájában alkalmazzák is ezeket a lehetőségeket, a kormányok igen komoly pénzügyi támogatásával.

Az országok egy másik részében — sajnos mi is ide tartozunk — még nem született meg az elhatározás, hogy egy mindenki számára elérhető, nyílt tömegképzési rendszert létrehozzanak.

Az őskor élővilágából a *homo sapiens* túl tudta élni az idők viharait, mert alkalmazkodni tudott a változásokhoz. A mai kor *homo sapiens cerebrus*-a a tanult ember, aki ismeri is és alkalmazza is a modern technológiát, és ezért képes lesz nemcsak túlélni, de önmaga hasznára is fordítani az elkövetkezendő évek nagy technikai-technológiai változásait.

KOVÁCS GYŐZŐ

A TC 3 elnöke: W. Brauer (NSZK)

A TC 3 munkacsoportjai

WG 3.1
Az informatika oktatása a középiskolában (1966)
Elnöke: P. Bollersler (Dánia)

WG 3.2
Fejlett oktatói rendszerek az információfeldolgozásban (1968)
Elnöke: W. F. Atchinson (USA)

WG 3.3
A számítógépek oktatási alkalmazása (1973)
Elnöke: R. Lewis (Nagy-Britannia)

WG 3.4
Középfok utáni és szakoktatás (1973)
Elnöke: B. Z. Barta

WG 3.5
Informatika az elemi iskolában (1984)
Elnöke: F. Louis (Nagy-Britannia)

WG 3.6
Távtanulás (1987)
Elnöke: Kovács Győző (Magyarország)

computer aided measuring & control

MÉRÉS ÉS IRÁNYÍTÁS SZÁMÍTÓGÉPPEL

III. RÉSZ

Cikksorozatunk folytatásaként egy programozható párhuzamos interfész alkalmazásával ismerkedünk meg. A választott gép, a C Plus/4, sokak kedvelt számítógépe az oktatásban és az otthonokban egyaránt. Bár erre a gépre sok jó program íródott már, a felhasználókban, barkácsolókban felmerül a kérdés, hogy játékon, kisebb adatfeldolgozáson és a tanuláson kívül mire lehetne még használni ezt a berendezést. Ehhez szeretnénk segítséget nyújtani ötletekkel és néhány konkrét megoldással.

Ehhez egy univerzális eszközt választottunk, az Intel 8255-öt. Ez az áramkör ugyan nem egészen kompatibilis a Commodore gépekben alkalmazott hardverelemekkel, de a Motorola hasonló funkciójú áramkörének beszerzése szinte lehetetlen.

Az áramkör belső felépítése és a lábkiosztása az *1. ábrán* látható. Az ábrából kitűnik, hogy összesen 24 bemeneti, illetve kimeneti vonallal dolgozhatunk. Az IC minden be- és kimenete TTL kompatibilis. A be- és kimeneti vezetékeket két főcsoportra bonthatjuk: az A portra a C port felső négy bitjével és a B portra a C port alsó négy bitjével. A C port megfelezve négy-négy vonalra programozható, külön ki- és külön bemenetként, míg az A és B portok iránya csak egyszerre állítható a nyolc vonalra.

Az áramkör vezérlőjelei:

D0-D7 — adatbusz-csatlakozás

RESET — aktív H szintű, törli a belső regisztereket, a portokat bemeneti üzemmódba állítja

$\overline{CS}$ — aktív L szintű, az IC kiválasztására szolgál

A0, A1 — az áramkör regisztereinek kiválasztására szolgál

$\overline{RD}$ — aktív L szintű, az olvasást vezérli a 8255-ből

$\overline{WR}$ — aktív L szintű, az írást vezérli a 8255-be

A konkrét illesztő áramkör megvalósításánál ennek a két jelnek az előállítása okoz gondot, mivel a C Plus/4 rendszerében ez a két funkció egy vonalon valósul meg (R/$\overline{W}$).

A *2. ábrán* az IC regisztereinek címzését láthatjuk. Az áramkör a CR parancsregisztaren keresztül programozható. Ennek a regiszternek a segítségével négy különböző üzemmód állítható be.

A *3/a ábra* a bitbeállító üzemmód formátumát tartalmazza. Ezzel az üzemmóddal csak a C port bitjeit állíthatjuk. A *3/b ábra* a parancsregiszter beállításának módjait tartalmazza. A B5, B6 bitekkel lehet az üzemmódokat kiválasztani. Ø-ás üzemmód: ez a legegyszerűbb üzemmód. A parancsregiszter többi bitjével (BØ-B4) állítható négy vezetékcsoport, a PAØ—7, PC4—7, PCØ—3 és a PBØ—7 be-, illetve kimeneti állapotba.

Az 1-es üzemmód: „kézfogás"-os (handshake) jellegű adatcsere valósítható meg az A vagy B port felhasználásával. A C port bitjeiből képződnek a vezérlőjelek. Ebben az üzemmódban lehetőség van megszakításkéréses működésre is.

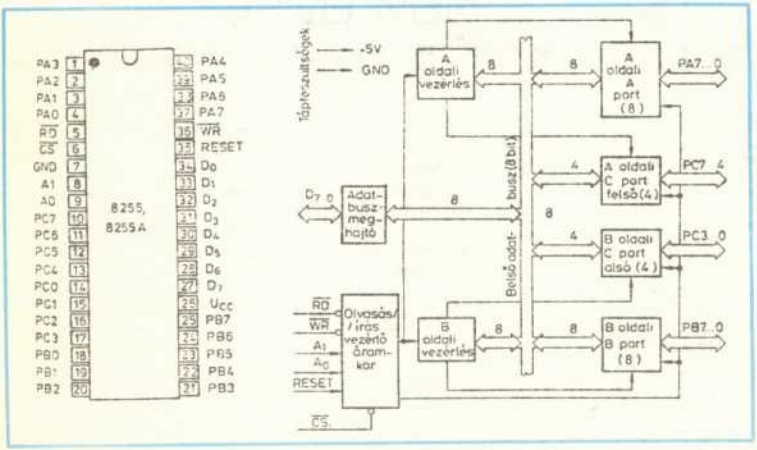
A *4/a ábra* kimeneti, a *4/b ábra* bemeneti módban ábrázolja a C port bitjeinek funkcióit (tulajdonképpen ezek az adatforgalmi vezérlőjelek).

A 2-es üzemmód: csak az A port üzemelhet ebben az üzemmódban. Ez a mód lehetőséget ad kétirányú adatforgalomra öt vezérlőjel segítségével (a vezérlőjelek a C port bitjei). Ilyenkor a B port Ø-ás módnak megfelelően programozható. Az *5. ábra* tartalmazza a 2-es üzemmódban a C port bitjeinek jelentését. A *6. ábrán* látható az az áramkör, amelyet a C Plus/4 bővítő kimenetére lehet csatlakoztatni.

Az áramkör működése

A NAND kapukból és inverterekből kialakított hálózat állítja elő a 8255 $\overline{CS}$ vezérlőjelét. Ez a cím $FD4Ø. A $FD4Ø az A port, a $FD41 a B

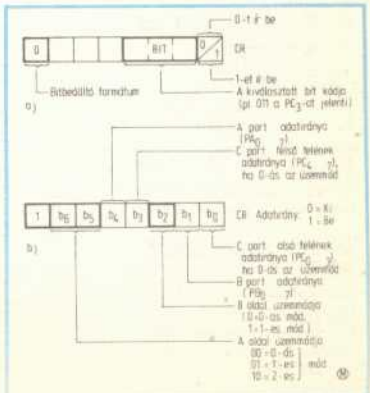
1. ábra

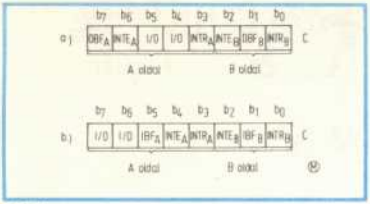


2. ábra

$\overline{CS}$	A_1	A_0		$\overline{RD}$	$\overline{WR}$		$\overline{RD}$	$\overline{WR}$	
0	0	0	PA port	0	1	BE	1	0	KI
0	0	1	PB port	0	1	BE	1	0	KI
0	1	0	PC port	0	1	BE	1	0	KI
0	1	1	CR parancsregiszter	—	—		1	0	beírás
1	X	X	adatbuszról leválasztva						

3. ábra





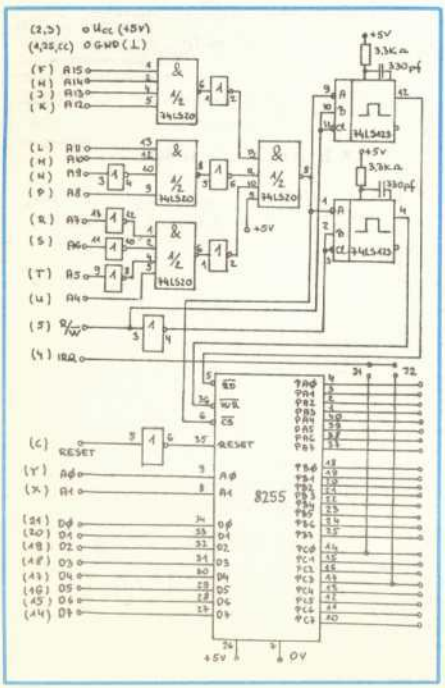
4. ábra

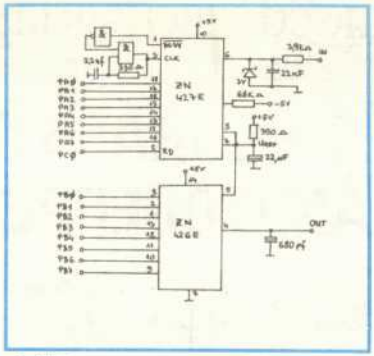
5. ábra

port, a $FD42 a C port, a $FD43 a CR parancsregisztere. A C Plus/4-nél a $FD40–$FDD0-ig terjedő terület nincs semmilyen tárterületre sem kikódolva. Ide tehát külső egységeket lehet elhelyezni. A 74LS123 a már említett R/$\overline{W}$ jelből állít elő megfelelő $\overline{R}$ és $\overline{W}$ jeleket a 8255 számára. A bemeneten zárójelben jelölt betűk és számok a C Plus/4 bővítő csatlakozójának jelzései.

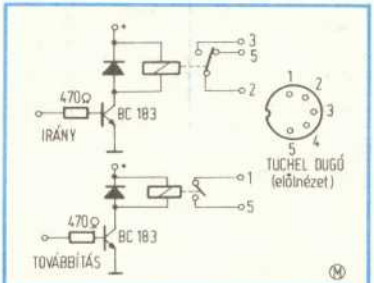
A 8255 programozásával kapcsolatban: BASIC-ből POKE utasítással lehet az áramkörbe írni, PEEK utasítással olvasni, és természetesen gépi kódból a megfelelő regiszterek programozásával lehet az IC-t vezérelni. A programozásnál ügyeljünk, hogy először az üzemmódot írjuk be, majd a megfelelő port címzésével adatot irhatunk vagy olvashatunk.

6. ábra

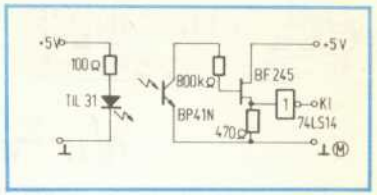




7. ábra



9. ábra



8. ábra

Néhány alkalmazási ötlet

A *7. ábrán* egy analóg ki- és bemenet áramköri megvalósítását láthatjuk. Ehhez megfelelő osztó és erősítő áramköröket kapcsolva, mérőműszereket, motorvezérlőket és szabályozóegységeket készíthetünk. A *8. ábrán* egy egyszerű infrakapu látható. Körülbelül 20 cm távolságig működik. Több fénykapu felhasználásával pontos időmérésre is alkalmazható (az oktatásban például fizikai méréseknél jól használható).

A *9. ábra* egy egyszerű ASPECTOMAT diavetítő-vezérlőt ábrázol. Ezzel az áramkörrel programozhatóan vezérelhetünk diavetítőt (felhasználása például az oktatásban: grafikus, bonyolult ábrák helyettesítése diaképekkel). A 8255 felhasználásával kiterjeszthetjük gépünk tárkapacitását is. Készíthetünk különféle EPROM-okhoz égetőt (égető impulzusokat előállító áramkörrel lehet kiegészíteni a kapcsolást).

Az építésnél, felhasználásnál mindenképpen ügyeljünk arra, hogy csak kikapcsolt állapotban csatlakoztassunk bármit is a géphez. A gép tápegysége a közölt kapcsolásokat még kiszolgálhatja, de nagyobb bővítésekhez külön tápegységet kell készíteni.

NIEBERL JÓZSEF

NEM EMBER-, KÉPERNYŐMENTÉS!

Ezzel a programmal a TV-Computer teljes képernyőjét kimenthetjük, eltárolhatjuk, majd alkalmas helyen visszatölthetjük. Menteni az EXT 0-val, visszahívni az EXT 1 utasítással lehet.

A 4010-es sorban foglalunk helyet a tárolandó képnek. A 4020-as sorban állítjuk be a mentő, töltő gépi rutinok címeit (6639, 6665).

Ha nincs szükség a teljes képernyő mentésére, a DATA listán az alábbi címeket kell módositani: 0,128 a videomemória kezdete, 35,26 a tárolás címe, 0,60 a mentett bájtok száma.

PÉNZES ÁKOS

```
4000 REM *** KÉPERNYŐ MENTŐ RUTIN ***
4010 LOMEM 22051: POKE 5920,35: POKE 5921,86
4020 POKE 33,239: POKE 34,25: POKE 35,9: POKE 36,26
4030 FOR I=6639 TO 6690: READ A: POKE I,A: NEXT: END
4040 DATA 62,80,50,3,0,211,2
4050 DATA 33,0,128,17,35,26,1,0,60,237,176
4060 DATA 62,112,50,3,0,211,2,201
4070 DATA 62,80,50,3,0,211,2
4080 DATA 33,35,26,17,0,128,1,0,60,237,176
4090 DATA 62,112,50,3,0,211,2,201
```

FELADATOK —MEGOLDÁSOK

```
;-------------------------------------
;       40 JEGYU SZAMOK SZORZASA
;
;     PINTER GABOR ,   1988.04.25.
;-------------------------------------

PROGRAM SEGMENT
        ASSUME  CS:PROGRAM
        ASSUME  DS:PROGRAM
        ASSUME  SS:PROGRAM
        ASSUME  ES:PROGRAM


; RENDSZERHIVAS
DOS     EQU     21H
; KILEPES
CONEXIT EQU     20H

; STRING BEOLVASAS
CONIST  EQU     0AH
; STRING KIIRASA
CONOST  EQU     9

; KOCSI VISSZA
CR      EQU     0DH
; SOREMENES
LF      EQU     0AH


; SZAM HOSSZA
; KARAKTEREKBEN
SZ      EQU     40
; MEMORIASZAVAKBAN
N       EQU     10

; SZAMOK FORRATURA:
; N ILL. 2N SZOM BINARISAN A
; SZAM ABSZOLUT ERTEKE,
;    (ALCSONYABB CIMEN
;     AZ ALACSONYABB HELYIERTEK)
; 1 BYTE-ON AZ ELOJEL.

; ILYEN EGY 40 JEGYU SZAM
SZAB40  STRUC
ELOJ40  DB      '+'
SZ40    DW      N DUP (?)
SZAB40  ENDS

; ILYEN EGY 80 JEGYU SZAM
SZAB80  STRUC
ELOJ80  DB      '+'
SZ80    DW      2*N DUP (?)
SZAB80  ENDS


        ORG     100H
START:  JMP     START2

; SZORZANDO
SZORZA  SZAB40  <>
; SZORZO
SZORZO  SZAB40  <>

; SZABOLASHOZ
MUNKA   DW      N DUP (?)

; AZ EREDMENY
EREDB   SZAB80  <>

; BEOLVASASI, KIIRASI PUFFER
; MERETE
PUFFIR  DB      SZ+2
; A TENYLEGESEN BEOLVASOTT
; KARAKTEREK SZAMA
        DB      (?)
PUFFER  DB      2*SZ+2 DUP (?)
; KIIRASNAL SOREMENES IS KELL
PUFFV:  DB      CR,LF,'$'

; SZOVEGEK:
RSZORZA DB      CR,LF
        DB      'Hegyvenjegyu'
        DB      ' szamok szorzasa'
        DB      CR,LF,CR,LF
        DB      'Iird be a szorzandot: '

                1
```

```
        DB      '$'
RSZORZO DB      CR,LF
        DB      'Iird be a szorzot: $'
REREDR  DB      CR,LF
        DB      'Az eredmeny:'
CRLF    DB      CR,LF
        DB      '$'



BEOLV   PROC    ; READ [DI]

; BEOLVAS EGY 40 JEGYU
; ELOJELES DECIMALIS SZAMOT.
; A BEOLVASOTT SZAMOT [DI]-TOL
; N SZOM HELYEZI EL.
; AZ ELOJEL [DI-1]-BE KERUL.


; REGISZTEREK MENTESE
        PUSHF
        PUSH    SI
        PUSH    AX
        PUSH    CX
        PUSH    DX

; A SZAM TORLESE
        CLD
        PUSH    DI
        MOV     CX, N
        XOR     AX, AX
  REP   STOSW
        POP     DI

; EGY SOR BEOLVASASA PUFFER-BA.
BESOR:  MOV     DX, OFFSET PUFFIR
        MOV     AH, CONIST
        INT     DOS

; A BEOLVASOTT KARAKTEREK VIZSGALATA
        MOV     SI, OFFSET PUFFER
        XOR     AH, AH
BE0:    LODSB
; URES SOR?
        CMP     AL, CR
        JNE     BEMSOR
BEUSOR: MOV     DX, OFFSET CRLF
        MOV     AH, CONOST
        INT     DOS
        JMP     SHORT BESOR
; SZAM KEZDODHET:
;   - ELOJELLEL
;   - SZAMJEGGYEL
BEMSOR: CMP     AL, '-'
        JE      BE1
        CMP     AL, '+'
        JE      BE1
        CMP     AL, '0'
        JB      BE0
        CMP     AL, '9'
        JA      BE0
; NINCS ELOJEL
        MOV     BYTE PTR [DI-1], '+'
        JMP     SHORT BESZAM

; ELOJELLEL KEZDODOTT
BE1:    MOV     [DI-1], AL
; AZ ELSO SZAMOT ELLENORZNI
BE1:    LODSB
; SOR VEGE?
        CMP     AL, CR
        JE      BEUSOR
        CMP     AL, '0'
        JB      BE1
        CMP     AL, '9'
        JA      BE1

; BEGYAN AZ ELSO SZAMJEGY
BESZAM: SUB     AL, '0'
        PUSH    AX
        PUSH    SI
; A REGI SZAMOT TIZZEL REGSZOROZZA
        MOV     AX, 10
        MOV     SI, DI
        CALL    SZOR

                2
```

```
; ES HOZZAADJA AZ UJ SZAMJEGYET
        POP     SI
        POP     AX
        ADD     [DI], AX
        JNC     BEHATV
; ATVITEL KELETKEZETT
        PUSH    DI
BEATV:  INC     DI
        INC     DI
        ADC     WORD PTR [DI], 0
        JC      BEATV
        POP     DI
BEHATV:
; KOVETKEZO KARAKTER VIZSGALATA
        LODSB
        CMP     AL, '0'
        JB      BESZV
        CMP     AL, '9'
        JBE     BESZAM

; SZAM VEGE
BESZV:  POP     DX
        POP     CX
        POP     AX
        POP     SI
        POPF
        RET

BEOLV   ENDP



SZOR    PROC    ; [DI] := AX * [SI]

; REGSZOROZZA AZ AX-BEN MEGADOTT SZABOT
; AZ [SI]-TOL KEZDODO N SZAVAS SZABBAL.
; AZ EREDMENY [DI]-TOL MAX. N SZO.

; REGISZTEREK ELMENTESE
        PUSHF
        PUSH    SI
        PUSH    DI
        PUSH    BX
        PUSH    CX
        PUSH    DX
        PUSH    AX

; SZORZAS AZ ALSO JEGGYEL
; EREDMENY DX:AX
        MUL     WORD PTR [SI]
; EREDMENY ELSO JEGY
        STOSW
; A TOBBI JEGGYEL VALO SZORZAS CIKLUSBAN
        MOV     CX, N-1
; A KELETKEZETT ATVITEL BX-BE
SZORC:  MOV     BX, DX
        POP     AX
        PUSH    AX
        INC     SI
        INC     SI
; SZORZAS A KOVETKEZO SZAMJEGGYEL
        MUL     WORD PTR [SI]
; ATVITEL HOZZAADASA
        ADD     AX, BX
; AZ OSSZEADAS SORAN IS
; KELETKEZHET ATVITEL
        JNC     SZORO
        INC     DX
SZORO:
; EREDMENY KOVETKEZO SZO KESZ
        STOSW
        LOOP    SZORC

; SZORZAS VEGE
        POP     AX
        POP     DX
        POP     CX
        POP     BX
        POP     DI
        POP     SI
        POPF
        RET

SZOR    ENDP

                3
```

Sorozatunkat elsősorban középiskolásoknak szánjuk, de reméljük, hogy minden olvasónknak tanulási lehetőséget és szórakozást nyújt.

A feladatok a Nemes Tihamér országos számítástechnikai verseny színvonalának felelnek meg. Minden esetben olyat választunk, amely röviden, gyorsan megoldható, de a megoldásboz ötletre van szükség. A megoldást mindig a következő számban közöljük.

Mivel változtatosságra törekszünk, különböző programozási nyelveket használunk. Az is előfordul majd, hogy egy feladatra több programnyelven is közlünk megoldást, ezzel is elősegítve az ismeretszerzést.

A szerkesztőség várja az olvasók, a versenyzők leveleit. A legötletesebb program beküldőjét könyvutalvánnyal jutalmazzuk. Ne feledjenek azonban a programhoz leirást is mellékelni!

5. FELADAT: SZORZÁS

Írjon programot, amely beolvas két negyvenjegyű előjeles egész számot, és kiirja azok szorzatát!

Megoldás

A feladat megoldható ugyan bármilyen programozási nyelven, de igazán gyors és jól használható akkor lesz, ha assemblerben irjuk meg. Ezért választottam én is ezt, még ha igy a program nehezebben érthető és hosszabb is lett. Az is célom, hogy az olvasót az assembler programozás technikájával megismertessem.

A mellékelt program IBM PC-n futott. A fordítást a MASM V5.0 assemblerrel végeztem. A 8086 assembler alkalmazása számos előnyt kinál: segiti a strukturált program irását, és szóhosszúságú számokat tud szorozni, ami a programot jelentősen egyszerüsiti.

Mielőtt a feladat megoldásához hozzáfognánk, el kell dönteni, hogyan ábrázoljuk a beolvasott számokat. Ábrázolhatunk egy bájton egy decimális számjegyet; használhatunk BCD-ábrázolást, vagy tárolhatjuk a számot bináris alakban is.

Mindegyik módszernek megvan a maga előnye, de a hátránya is. Ha bináris alakban tároljuk a számot, akkor a beolvasásnál és az eredmény kiirásánál azt át kell konvertálni, de cserébe a számitás egyszerűbb és gyorsabb lesz. Ha viszont valamilyen decimális ábrá-

zolást választunk, akkor a beolvasásnál és kiírásnál lesz egyszerűbb dolgunk.

A közölt programban az előbbi módszert alkalmaztam. A szám alakja a következő. A szám első bájtja tartalmazza az előjelet, karakteres alakban (azaz „+" pozitív, „−" negatív szám esetén). Az előjel után a szám abszolút értéke 10 szón binárisan tárolódik. Alacsonyabb címre az alacsonyabb helyi érték kerül.

Mivel két negyvenjegyű szám szorzata nyolcvanjegyű, az eredmény tárolására szükség van egy ilyen nagy számra is. Ez a negyvenjegyűhöz hasonlóan, de kétszeres hosszon történik.

A szorzóalgoritmus megírásánál az általános iskolában megismert írásbeli szorzás lehet a példánk. Abban a szorzandót mindig csak a szorzó egy számjegyével kell megszoroznunk, majd a számítás utolsó lépésében adjuk össze a részletszorzatokat.

A szorzóalgoritmusban is így kell eljárnunk, azzal a különbséggel, hogy az összeadást célszerű minden egyes részletszorzat kiszámítása után elvégezni, és számjegy helyett mindig szót kell értenünk.

Ezek után nézzük a programot! Először a felhasznált konstansokat és struktúrákat definiáljuk. A szám hosszúságát is szimbólum jelöli: SZ a szám hosszúsága tízes számrendszerben, H az ábrázolásához szükséges szavak száma ($H > 0{,}2076 * SZ + 1$). Látható, hogy a program nemcsak 40, hanem SZ és H megfelelő megválasztásával tetszőleges hosszúságú számok szorzására is alkalmas.

Ezután a változóterület lefoglalása következik. A szorzandó, a szorzó és az eredmény részére lefoglalt területen kívül szükség van a részletszorzatok tárolásához, a beolvasáshoz és a kiíráshoz munkaterületre. A szövegkonstansok is itt találhatók.

A program láthatóan szubrutinokra tagolódik. Tekintsük át ezek feladatát és működését!

BEOLV végzi a negyvenjegyű számok beolvasását. Először a számot törli, és az operációs rendszer sorbeolvasó rutinjának segítségével beolvas egy sort. A sorban megkeresi az előjelet, ha van, majd a számjegyeket olvassa egymás után. Egy új számjegyet úgy illeszt a már meglevő számhoz, hogy a számot megszorozza tízzel, majd hozzáadja az új számjegyet.

SZOR egyszavas számmal szorozza meg a negyvenjegyű számot. Az első jeggyel való szorzást külön végzi, mert ekkor még nincs átvitel. A többi jeggyel a szorzás ciklikus. Az előző jegyen keletkezett átvitelt BX tárolja.

```
PLUSZ   PROC    ; [BX] := [BX] + [DI]

; HOZZAADJA AZ [DI]-TOL KEZDODO
; N SZAVAS SZABOT A [BX]-TOL KEZDODO
; SZAMHOZ. AZ EREDMENY [BX]-TOL.

; REGISZTEREK MENTESE
        PUSHF
        PUSH    SI
        PUSH    DI
        PUSH    AX
        PUSH    CX

; CIKLUS KEZDOERTEKEK
        CLD
        CLC
        MOV     SI, DI
        MOV     DI, BX
        MOV     CX, N
; AZ ELSO N SZO OSSZEADASA
PLC:    LODSW
        ADC     AX, [DI]
        STOSW
        LOOP    PLC

; ATVITEL
PLATV:  JNC     PLV
        ADC     WORD PTR [DI], 0
        STOSW
        JC      PLATV

; OSSZEADAS VEGE
PLV:    POP     CX
        POP     AX
        POP     DI
        POP     SI
        POPF
        RET

PLUSZ   ENDP



KIIR    PROC    ; WRITE [SI]

; KIIR EGY 80 JEGYU
; ELOJELES DECIMALIS SZAMOT.
; A SZAM [SI]-TOL,
; 2N SZOM HELYEZKEDIK EL.
; AZ ELOJEL [SI-1]-EN VAN.

; REGISZTEREK MENTESE
        PUSHF
        PUSH    SI
        PUSH    DI
        PUSH    AX
        PUSH    BX
        PUSH    CX
        PUSH    DX

; CSOKKENO CIMEK
        STD
; DI AZ UTOLSO KARAKTERRE MUTAT
        MOV     DI, SI
        ADD     DI, 4*N-1
; A KIMENETI PUFFER BEALLITASA
        MOV     BX, OFFSET PUFFV
        MOV     CX, 2*SZ+1
; EZZEL FOG OSZTANI
        MOV     BL, 10

; KULSO CIKLUS
KIKAR:  PUSH    DI
        PUSH    CX

; A 80 JEGYU SZAM OSZTASA TIZZEL
        XOR     AH, AH
        MOV     CX, 4*N
KI10:   MOV     AL, [DI]
        OR      AX, AX
; A NULLAT FOLOSLEGES TIZZEL OLOSZTANI
```



```
        JZ      KI0
        DIV     BL
; MARADEK: AL, HANYADEK: AH
KI0:    STOSB
; A MARADEKRA MEG SZUKSEG VAN
        LOOP    KI10

; AH-BAN AZ OSZTAS MARADEKA
        POP     CX
        POP     DI
        ADD     AH, '0'
; EGY SZAMJEGY KESZ
        DEC     BX
        MOV     [BX], AH
        LOOP    KIKAR

; AZ OSSZES SZAMJEGY KESZ
; VEZETO 0-K ELHAGYASA
KIV0:   CMP     BYTE PTR [BX], '0'
        JNE     KIVN
        INC     BX
        JMP     SHORT KIV0
; AZ EREDMENY 0?
KIVN:   CMP     BYTE PTR [BX], CH
        JNE     KIELOJ
        DEC     BX
        JMP     SHORT KISZAR
; ELOJEL KIIRASA
KIELOJ: CMP     BYTE PTR [SI-1], '-'
        JNE     KISZAR
        DEC     BX
        MOV     BYTE PTR [BX], '-'
; A TELJES SZAM KIIRASA
KISZAR: MOV     DX, BX
        MOV     AH, CONOST
        INT     DOS

; KIIRAS VEGE
        POP     DX
        POP     CX
        POP     BX
        POP     AX
        POP     DI
        POP     SI
        POPF
        RET

KIIR    ENDP


; FOPROGRAM
START?:
; ELSO SZAM
        MOV     DX, OFFSET RSZORZA
        MOV     AH, CONOST
        INT     DOS
        MOV     DI, OFFSET SZORZA.SZ40
        CALL    BEOLV

; MASODIK SZAM
        MOV     DX, OFFSET RSZORZO
        MOV     AH, CONOST
        INT     DOS
        MOV     DI, OFFSET SZORZO.SZ40
        CALL    BEOLV


; A SZORZAS

; EREDMENY FELSO JEGYEINEK TORLESE
        CLD
        MOV     DI, OFFSET EREDM.SZ80+N
        MOV     CX, N
        XOR     AX, AX
  REP   STOSW

; ELSO JEGGYEL SZORZAS
        MOV     SI, OFFSET SZORZO.SZ40
        LODSW
        XCHG    DX, SI
        MOV     SI, OFFSET SZORZA.SZ40
        MOV     DI, OFFSET EREDM.SZ80
        CALL    SZOR
```



```
; TOBBI JEGGYEL SZORZAS
        MOV     CX, N-1
        MOV     BX, DI
        MOV     DI, OFFSET MUNKA
SZOR40: INC     BX
        INC     BX
        XCHG    DX, SI
        LODSW
        XCHG    DX, SI
; SZORZAS
        CALL    SZOR
; OSSZEADAS
        CALL    PLUSZ
        LOOP    SZOR40

; ELOJEL MEGALLAPITASA
        MOV     [EREDM.ELOJ80], '+'
        MOV     AL, [SZORZO.ELOJ40]
        CMP     AL, [SZORZA.ELOJ40]
        JE      SZOREL
        MOV     [EREDM.ELOJ80], '-'
SZOREL:

; EREDMENY KIIRAS KOVETKEZIK
        MOV     DX, OFFSET EREDM
        MOV     AH, CONOST
        INT     DOS
        MOV     SI, OFFSET EREDM.SZ80
        CALL    KIIR


; KILEPES
        INT     CONEXIT

PROGRAM ENDS
        END     START
```

PLUSZ egy negyvenjegyű számot ad hozzá egy tetszőlegesen hosszúhoz. Az összeadás szavanként, ciklusban történik.

KIIR írja ki a nyolcvanjegyű előjeles számot. A kiíratásnál a bináris számot decimálissá kell alakítani.

Ez az euklideszi algoritmussal valósítható meg. A bináris számot elosztjuk tízzel, a maradék adja a tízes számrendszerbeli szám legutolsó jegyét. Ezután a hányadost osztjuk el tízzel, a maradék adja a tízes számrendszerbeli szám következő jegyét. Ezt ciklikusan ismételve, amíg a hányados nulla nem lesz, számíthatók a további jegyek.

KIIR-ban az osztandót és a hányadost ugyanazon a helyen tároljuk, ezért az algoritmus nehezebben látható.

A vezető nullák elnyomása és az előjel kiírása zárja a rutint.

A főprogramban van a negyvenjegyű számok szorzását elvégző rész. Ezt a beolvasást és a kiírást ellátó utasítások keretezik.

A szorzás a közönséges írásbeli szorzáshoz hasonló: az első jegygyel való szorzással létrejött részletszorzathoz adja hozzá a további részletszorzatokat. A szorzás utolsó lépése az eredmény előjelének megállapítása.

A 6. feladat a mostaninak éppen az inverze lesz.

6. FELADAT: OSZTÁS

Írjon programot, amely beolvas egy nyolcvan- és egy negyvenjegyű előjeles egész számot, majd kiírja azok hányadosát és a maradékot!

PINTÉR GÁBOR

MIÉRT EZT, S MIÉRT NEM MÁST?!

Kedves . . .! Írd meg, hogy milyen előzmények után, milyen meggondolásból döntöttél úgy, ezt az iskolát választod. Nem kell sokat írnod, de az legyen feltétlenül őszinte.

Ilyen tartalmú leveleket küldtünk a Hámán Kató Szakközépiskola leendő első osztályos diákjainak, kutatva ezzel azt, hogy mitől és miért olyan vonzó a programozás mint szakma, milyen késztetések hatására szánják el magukat a számítástechnika tanulására.

Ami meglepett, a levelek nagy többségére — kettő híján az összesre — válasz érkezett, pedig nem volt kötelező. Úgy vélem, ez már önmagában is az elhatározás komolyságáról tanúskodik, mert hiszen mindegyikük, egyik a másikról mit sem tudva, fontosnak tartotta a jeladást. Nem valószínű, hogy bármelyik hagyományos szakmára fejét adó tizenéves ne legyintett volna: nem lényeges, emiatt vagy amiatt és kész, minek írjak?!

Ami a lényeget illeti: sorra veszem a levelek legérdekesebb, legjellemzőbb azonosságait, vonásait. — Egyre nyilvánvalóbb, hogy akinek jók a tanulmányi eredményei, az nem feltétlenül gimnáziumba jelentkezik. — „Tanulmányi eredményeim alapján fölvettek volna egy jó nevű gimnáziumba, de én jobban szeretném, ha a négy év elvégzése után szakma lenne a kezemben." Vagy: — „A bizonyítványom a 7. osztály végén nagyon jó volt, ezért javasolták, hogy menjek gimnáziumba. Nekem ehhez nem sok kedvem volt, mert ha érettségi után a felvételi nem sikerül, akkor nagyon nehéz elhelyezkedni." — Való igaz, egy „nagy jövő előtt álló" — ahogy sokan fogalmaztak — szakmával a kezében könnyebben helyezkedhet el bárki, mint egy gimnáziumi érettségivel.

Hol kóstoltak bele először a számítástechnikába, avagy szereztek tudomást róla? — Legtöbbjüket szakkörök, iskolai fakultációk indították útjukra. Ami jelzi azt is, milyen nélkülözhetetlen a korai csipegetés, csemegézés. Sajnálatosnál is több viszont, hogy egy kivétellel semelyikük sem említette a rádiót, televíziót és az újságokat (a magazint sem) mint ösztönzést. — „Hatodikos voltam, amikor megismerkedtem a számítástechnikával a kerületi úttörőházban. Egyre jobban kezdett érdekelni, és nyolcadikos koromban a pályaválasztáskor úgy döntöttem, ezt választom." — Az ilyen sorok folytatása mögött érződött, jó néhányan abban a tévhitben élnek, hogy ugyanaz következik az iskolában, amit a szakkörökön elkezdtek, pedig ez csak részben van így. A Basic nevű tantárgy tényleg fedi az elképzeléseiket, de a programozás sokkal többet takar, mint amit a kisgépek nyújthatnak a programozójuknak.

Nem hanyagolható el egy másik csoportja sem az elsősöknek, akik a családban szívták magukba az olthatatlan vágyat. — „Választásomhoz hozzájárult az is, hogy családunkban többen — így szüleim is — számítástechnikával foglalkoznak." — Természetesen a szülőktől kapott gép is, az iménti ekhez hasonlóan, sokuknál meghatározó volt. Végül azok vannak kisebbségben, akik először csak azt szeretnék megtudni, mi fán terem, és azután akarják elsajátítani a programozást. — „Mivel a számítástechnikának távlatai vannak, és számomra izgalmasnak tűnik a bővebb megismerése, úgy gondoltam, erre irányítom az érdeklődésemet." — Persze, ha az ilyen érdeklődés később sem csökken, egy év elmúltával lefaragható a beavatottabbakkal szembeni lemaradás.

„Az iskolában a tantárgyak közül a matematikát kedveltem a legjobban. Szüleim javasolták, hogy olyan pályát válasszak, ami ezzel a tárggyal összefügg, kapcsolatban van." — A programozás tényleg kapcsolódik a matematikához, hiszen mit sem érne a technika a matek és a fizika nélkül. De tévedés azt hinni, hogy aki szereti a matematikát, az megszereti a programozást is. Ehhez más is kell. Ha valaki ismeri a gép összes utasítását és azok használatát, korántsem biztos, hogy programokat is tud írni. Sajnos nagyon sok ember szemében a számítástechnika még megmaradt a XX. század alkimiájának. Misztikus, csodákat művelő gépnek, ami az emberek helyett gondolkodik. Pedig éppen ez utóbbi az a más, ami a programíráshoz kell.

Végül minden mostani és jövőbeni elsősnek érdemes még egy levélből idézni. — „Azt hiszem, hogy engem a pályára a számítógép szeretete és az alkotás, alkotni akarás vágya vezérelt. Annál jobb érzés nem sok van, mint amikor az ember egy új, saját készítésű program utolsó sorait írja, és munkájának eredménye megjelenik a képernyőn." — Ezekhez a gondolatokhoz nem kell hozzáfűzni semmit.

Még csak annyit: a mintegy két tucatnyi levél alapján túlzás lenne azt állítani, hogy az ország minden, ezt a szakmát választó elsősére általánosíthatóak az iménti motivációk, befolyásoló körülmények. Ennek ellenére mégis úgy tűnik, a cseppben felismerhető ez alkalommal is a tenger.

VÁMOS SÁNDOR

ATTR ABLAK, SZÍNVÁLTÓ RUTIN ZX-SPECTRUMRA

Egy program befejezésekor a szerző arra törekszik, hogy műve minél szebb legyen. Ez a gépi kódú rutin segít ebben. A képernyőre írás után a villogó, mozgó ATTR színesebbé teheti a programot és emelheti színvonalát. A képernyőre írás után pozicionálhatjuk — oszlop és sor beírásával — az ATTR ablak kezdő és záró koordinátáját úgy, mint ahogy a PRINT AT „BASIC" utasításnál szoktuk. Ezt a következőképpen végezhetjük el.

A 23296-os memóriacímre írjuk az ablak bal felső sarkának oszlopát, a 23297-es címre pedig sorszámát. Ugyanígy készítjük el az ablak jobb alsó sarkát is. Ezt a pozicionálást a 23298 (oszlop) és a 23299 (sorszám) memóriacím átírásával hajthatjuk végre.

A rutin indítása a BASIC betöltőprogram *(1. lista)* után RANDOMIZE USR 30000. Ekkor egy ATTR-okból álló mozgó négyszöget láthatunk. Az ATTR-ok mozgása, tehát a rutin addig működik, amíg bármilyen billentyűt le nem nyomunk.

A rutin szemléltetésére írjuk be a *2. listán* látható demo programot, miután a BASIC betöltővel végeztünk.

PUSKÁS LÁSZLÓ

1. lista

```
10 LET M=0
20 READ A: IF A<0 THEN STOP
30 POKE 30000+M,A
40 LET M=M+1:GOTO 20
50 DATA 175,50,8,92,205,66,117,118,118,
   118,58,8,92,254,0,40,243,201,58,1,91,
   71,58,0,91,205,186,117,34,7,91,17,1,
   0,58,0,91,79,58,3,91,145,50,5,91,71,
   205,159,117,58,1,91,71,58,2,91,205,
   186,117,34,9,91,17,1,0,58,5,91,60,71,
   205,159,117,42,7,91,17,32,0,58,0,91,
   79,58,2,91,145,50,6,91,71,205,159,
   117,58,3,91,71,58,0,91,205,186,117,
   17,32,0,58,6,91,71,58,4,91,79,203,63,
   203,63,119,121,198,32,50,4,91,25,16,
   238,58,4,91,222,32,50,4,91,201,111,
   38,0,41,41,41,41,41,35,16,253,17,0,
   88,25,201,-1
```

2. lista

```
10 PRINT AT 9,4;INK 2;"A PROGRAMOT KESZ
   ITETTE :";AT 10,11;BRIGHT 1;PAPER 6;
   "PUSHI SOFT";AT 11,14:BRIGHT 0;"1987"
20 POKE 23296,7:POKE 23297,3:POKE 23298,
   12:POKE 23299,28
30 RANDOMIZE USR 30000
```

SZINTE EGY LÉLEGZETRE...

Azt hiszem, nem tévedek nagyot: a programirás alkalmasint egyik legfőbb erénye a tömörség és a rövidség. Természetesen hosszú program is lehet nagyon jó, ha a dolog is úgy kívánja. Ha viszont nyújtja valaki, mint a rétestésztát!... TVC-re irt grafikai trükkjeimet, hatásaimat éppen ezt elkerülendő készitettem. Igyekeztem programtyűimet minél rövidebbre fogni, hogy mindegyik szinte egy lélegzetre beírható legyen. A miniprogramoktól persze senki se várjon csodát, de arra alkalmasak, hogy bővítsék a gép egyelőre még vékonyka „szótárát".

Az első egy érdekes, szines — VIDEOTON — feliratot ir ki a képernyőre *(1. lista)*. Ez csak egy fajta, de próbálgatással sok szép betűtipust el lehet érni, így karakterek tervezgetése nélkül is látványossá tehetjük programunkat.

Ha közvetlenül a videoportra küldjük a sztringünket, (PRINT 0:"krixkrax"), akkor az az aktuális rajzolási pozíciótól kezdve, az aktuális színben, az aktuális keresztezési mód szerint fog megjelenni.

A 2896-os bájt első két bitje állitja be a karakter-felülirás módját. (1234) 2 esetén csak a tintaszinű pontokat irja a gép.

Ilyen módszerrel nagyon egyszerűen térbeli betűket is lehet rajzolni, úgy hogy a kiirt karaktersort még egyszer kiírjuk, függőlegesen és vizszintesen négy-négy egységgel eltolva, más szinnel. (A 2896-ra ne felejtsünk 2-őt irni!) Ezt is nehezebb volt elmondani, mint megcsinálni . . .

A második is egy egyszerű BASIC program. Az általam futófénynek nevezett effektust mutatja be: a *2. lista* beírásával fogaskerekeket látunk forogni.

A fogaskerekeknek a „fogait" sorra a következő szinnel rajzoljuk. (1231231 . . .) A futófény, vagyis itt a forgás úgy jön létre, hogy a palettán az összes szint háttérszínüre tesszük. Mégpedig úgy, hogy kivesszük az egyest, a kettest majd a hármast. A három szín egyértelmű forgásirányt ad meg. (2-nél hol erre, hol arra látjuk forogni.) A mozgás három fázisát bizonyos késleltetéssel váltogatjuk, és már forog is a fogaskerék, a kőrhinta, fut a futófény, vagy rugdoshatja a szódás a lovát. Csupán a mozgás három fázisát kell külön-külön szinnel megrajzolni.

Ezúttal végül az LDIR utasitás BASIC-ből való felhasználását mutatom be *(3. lista)*. Ezt a példa arra használja, hogy a képernyőt (csak 16200 bájt) pillanatok alatt elmentse, majd visszatöltse. Az LDIR blokkmozgatást végez. Tehát például áthelyezhetek vele a 6639-es címről a 6700-ra tiz bájtot. A rutin betöltése után az LDIR-t az EXT-tel érjük el BASIC-ből. EXT 0, forráscím, célcím, áthelyezendő bájtok száma.

```
ORG 6639: LOAD 6639
DI                      ;megszakitás tiltás
LD A,BO:OUT (2),A       ;video belapozás
LDIR                    ;blokkmozgatás
EI:RET                  ;megszakitásengedélyezés, BASIC
END
```

BORDÁS BENCE

1. lista

```
10 REM **** 1. ****
20 GRAPHICS 4
30 POKE 2896,2
40 A$="VIDEOTON"
50 SET MODE 3
60 SET INK 0:PLOT 330,810:SET INK 3:PRINT #0:A$
70 SET INK 0:PLOT 338,818:SET INK 2:PRINT #0:A$
80 GET
```

2. lista

```
10 REM **** 2. ****
20 GRAPHICS 4
30 FOR I=0 TO 2*PI STEP PI/18
40 PLOT 512,480;512+200*SIN(I),480+200*COS(I)
45 PLOT 312,680;312+200*COS(I),680+200*SIN(I)
50 J=J+1:IF J>3 THEN J=1
60 SET INK J
70 NEXT I
80 SET PALETTE 0,85,0,0
85 FOR J=0 TO 30:NEXTJ
90 SET PALETTE 0,0,85,0
95 FOR J=0 TO 30:NEXTJ
100 SET PALETTE 0,0,0,85
105 FOR J=0 TO 30:NEXTJ
110 GOTO 80
```

3. lista

```
10 REM **** 3. ****
20 GRAPHICS 4
30 LOMEM 24700
40 POKE 33,239:POKE 34,25
50 READ A:IF A=555 THEN 100
60 POKE 6639+B,A:B=B+1:GOTO50
70 DATA 243,62,80,211,2,237,176,251,201,555
100 REM DEMO
110 LIST
120 PRINT:PRINT"GOMBNYOMASRA ELTESZI...":GET
125 EXT 0,32767,6700,16200
130 CLS:PRINT"GOMBNYOMASRA VISSZA!"
140 GET
150 EXT 0,6700,32767,16200
```

ÖRÖKÉLET

Úgy látszik, Tuba Imre kifogyhatatlan az örökélet-kódokból. Mostani gyüjteménye C16-os és C Plus/4-re bővitett C16-os gépen működik.

3-D QUASARS	7198,173
AIRWOLF 2	7962,165
ARENA	4794,0
AURIGA	8113,189
BOMB JACK	5196,173
BOOTY	9013,165
BURGER HOUSE	8046,173
CAVE FIGHTER	4922,172
idő	4309,234
	4310,234
	4311,234
CRAZY GOLF	4174,201
	6890,0
Ezután E gombbal váltható a pálya.	
CREATURE CRAWL	10676,173
DIAGON	9606,0
DIAMOND MINE 1	5938,173
	7370,0
	8251,0
	9224,0
DIAMOND MINE 2	4206,173
GHOSTS 'N' GOBLINS	4365,173
	4393,165
GUNLAW	4627,173
HEKTIK	6468,173
	7363,173
JETBRIX	8864,173
time	16055,169
	16056,28
	16057,234
	8369,29
KIKSTART 1-3.	10403,165
idő	9033,189
	9036,189
	9082,189
	9085,189
	11938,12
	11946,32
KILLAPEDE	10496,173
KING OF KINGS	9725,173
	9747,173
KNOCK OUT	4630,173
L 'UMO	8960,173
LASER 900	7362,173
LASER ZONE	5426,100
elektro	5671,101
MAIL TRAIL	6613,173
days	6159,189

TANULÁS — SIKER — TÁBORTŰZ

Az idén is — nemzetközi részvétellel először — megrendezték a Szovjetunióban azt a majd háromhetes számítástechnikai találkozót, melynek színhelye a Novoszibirszk melletti Recskunovka faluban a Szibirjak ifjúsági tábor volt. A rendezők több országba is küldtek meghívót. Bulgáriából, az NDK-ból és Magyarországról érkeztek kísérőikkel a négytagú csapatok. Itthoni berkekből az NJSZT állította össze a magyar „képviselők" stábját. A magyar színeket Biczó Tibor és Farkas Károly zalaegerszegi, valamint Futó Tibor és Szoldán Péter budapesti középiskolások képviselték.

Repülőutunk szép volt, de fárasztó. A határás időeltolódással együtt, hajnali négykor értünk a táborba, ahol kétágyas szobákban szállásolták el a társaságot. Igaz, a szobák elég szűkek voltak, s a meleg vizet is csak néhány nap múlva élvezhettük, de megismerkedhettünk az orosz konyha eddig csak hallomásból ismert ízeivel.
A tábor eredeti célja — amint azt orosz neve, a Létnyaja Skola is mutatta — a tanulás volt. Ezért sok előadást hallgattunk végig, ám hiányos orosz tudásunk miatt csak bizonyos dolgokat értettünk meg ezekből. Viszont három versenyen is részt vettünk, olyanokon, ahol számítástechnikai tudásunk megméretett.

Mindjárt az érkezés napján nyolc feladatot kaptunk: algoritmusokat kellett minél rövidebben megírni. A megoldásokat egy héten belül kellett megadnunk. A munkák eredményét az utolsó napon hirdették ki, vagyis azt, hogy melyik csapat készített jobb, szebb, rövidebb vagy gyorsabb algoritmusokat. Büszkén mondhatom: a magyar „küldöttség" nyerte meg ezt a tusát.

Hamarosan sor került a második erőpróbára is. A résztvevő országoknak egy-egy kétversenyzős csapatot kellett kiállítaniuk. Mi hosszas vita, egyezkedés után Futó Tibort és Szoldán Pétert „delegáltuk". A feladat lényege az volt, hogy a csapat két tagja ugyanarra a témára ír egy-egy programot, de különböző helyiségekben, és az elkészített programot nem volt szabad gépen futtatni. Illetve, amikor mindkét versenyző befejezte munkáját, akkor helyet cseréltek, s a másik programját belőtték, vagyis használhatóvá tették. BASIC vagy Turbo Pascal nyelven lehetett dolgozni. Csapatunk a BASIC-et választotta, s nyert ebben a versenyszámban is.

A következő feladványt oldottuk meg. — Vegyünk egy n lapból álló kártyacsomagot, s a kártyák egyik oldalára írjuk rá a természetes számokat 1-től n-ig. Ezután a kártyákat keverjük össze, majd a paklit fordítsuk meg. Ekkor írjuk a lapok eme oldalára is a természetes számokat 1-től n-ig. Ismét keverjük meg a kártyacsomagot, de most már a keverés közben egyes kártyákat megfordítunk. Ezután a lapokat letesszük egymás mellé az asztalra. Írjunk tehát olyan programot, amely n és a kártyákon felül és alul található számok ismeretében megadja, melyik kártyákat kell megfordítani ahhoz, hogy a felső oldalon valamennyi szám pontosan egyszer szerepeljen!
A harmadik versenyben nem jeleskedtünk. A feladat az lett volna, hogy a megérkezésünkkor kiválasztott témára — Júlia, Mandelbrot-halmaz, fraktálok stb. ábrázolása — írt programról angolul tartsunk előadást. Sajnos, csapatunkat lázzal fűszerezett torokgyulladás kapta el, így nem tudtunk részt venni a megméretésen. Itt egyébként az irkutszkiak csapata győzött.

Vendéglátóink a számítástechnikához kapcsolódó programokon kívül még sok mindennel gondoskodtak okosításunkról és szórakoztatásunkról. Megmutatták például a Novoszibirszki Atomfizikai Kutató Intézetet, ellátogattunk a városi képtárba, s gyönyörködhettünk a ragyogó gyűjteményben. És megtekintettünk egy vízierőművet is. Jártunk a helyi történeti múzeumban, és végignézhettük a néhány esztendeje elhunyt, világhírű szovjet előadóművész és színész, Vigotszkij emlékét őrző kiállítást. Természetesen számos műemléket is felkerestünk.
Rockkoncerten is voltunk. A helybélieknek még újdonság az ilyesfajta rendezvény, noha a glasznoszty térhódítása óta mind gyakrabban lépnek fel rockegyüttesek. A fiatalok tisztes számban látogatják ezeket a műsorokat, noha esténként videodiszkó is van a szórakozóhelyeken.

A búcsú hetében a Novoszibirszki Számítástechnikai Klub vendégszeretetét élvezhettük a város legjobb, legelegánsabb szállodájában. Egyikünk sem panaszkodhatott: az ellátás valóban magas színvonalú volt.
Utolsó ott töltött éjszakánkon még egy gazdag programról gondoskodtak vendéglátóink: az ott szokásos, sőt hagyományos tábortűz mellett köszönhettünk el Novoszibirszktől és a három hét alatt szerzett, megszeretett barátainktól.
Visszatekintve milyennek ítéljük meg a nyári számítástechnikai tábort? Bizton állíthatjuk, hogy hasznos volt. Sok mindent tanultunk, láttunk, tapasztaltunk. Például megismerhettük a Szovjetunió egyik tudományos központját, bepillantást kaptunk a szovjet fiatalok életébe, feltérképezhettük korosztályunk számítástechnikai tudását. És nem utolsósorban, három héten át jól éreztük magunkat.
Az NJSZT pedig jövőre ismét meghívást kapott . . .

SZOLDÁN PÉTER

RENEW C16-ra

Úgy hiszem, ez a legrövidebb RENEW, amit eddig láttam. Mindössze négy sor *(1. lista)*. Hívása: SYS $cim.

Az ERASE a SYS után megadott tömböket törli úgy, hogy a mögötte álló tömböket felcsúsztatja és a mutatókat átállítja. Az a kevés magyarázat, ami ennek megértéséhez kell, a *2. listán* megtalálható. Talán még annyit, hogy a $96A5-ön levő Address rutin a tömb címét $5F-$60-ba teszi, az elem címét pedig $47-$48-ba. Az ERASE rutin hívása: SYS $cim, tömbnév.

A *3. lista* olyan BASIC szubrutin, amely futás közbeni függvénydefiniálásra alkalmas. Előnye, hogy ameddig a memória tart, tetszés szerint sok függvényt definiálhatunk vele.

A *4. lista* ennek Assembler változata. A C16 bosszantóan kis memóriája miatt itt csak egy függvényt használhatunk. Hívása: SYS $cim, változó neve.

Címnek a kazettapuffer kezdetét adtam meg. A magyarázatok a listákon vannak. Sokkal szebb lenne a megoldás felhasználói tokenekkel, de ezt 16 k-val aligha érdemes megcsinálni, főleg ha a grafikus szolgáltatásokat is igénybe akarjuk venni. A legjobb, ha a FUNCTION rutint használó BASIC program első sora egy REM•••••••, így a •-ok helyére kerülhet a függvény szövege. Ez listázásnál nem a legszebb, de ezt esetleg le is tilthatjuk.

STEIR MIKLÓS

1. lista

```
RENEW:

cim:lda  o1           ;(&o332)
     tay
     sta (&2b),y
     jmp &8818
     ;hivasa: sys(cim)
```

2. lista

```
ERASE: jsr &o473          ;vezezo
 cim : jsr &96a5          ;(&o332),valtozo be
       ldy  o2            ;(&57)+2,a kov. tomb cime
       lda(&5f),y         ;kovetkezo tomb
       adc &5f
       sta &47            ;hivasa:sys(cim),tomb(dim)
       iny
       lda(&5f),y
       adc &6o
       sta &48
       ldy oo
       lda &47            ;(o346)
       cmp &31            ;a kov,tomb cime=a tombok vege?
       bne &o352          ;nem-ciklus
       lda &48            ;igen
       cmp &32
       beq &o365
       lda(&47),y
       sta(&5f),y
       inc &47            ;a mutatok allitasa
       bne &o35c
       inc &48
       inc &5f            ;(o35c)
       bne &o346
       inc &6o
       jmp &o346
       lda &5f            ;(o365)
       sta &31            ;tombvege mutato:=&(5f-6o)
       lda &6o
       sta &32
       rts                ;basichez
```

3. lista

```
BASIC FUNCTION IN:

1o gosub 1ooo
2o end
21
22   ;fenntartott
.
.
4o  ;foprogram
5o end
1ooo input"be a fv-t!";N
1o1o wait 198,6o
1o2o printchr$(72);chr$(13);chr$(13)
1o25 on N goto 1o7o,1o8o
1o3o fori=1 to 3o:printchr$(73):next i
1o4o return
1o7o ?"21 deffnf(x)=";chr$(13);"run 21";goto 1o3o
1o8o ?"22 deffng(x)3";chr$(13);"run 22";goto 1o3o
(A Z O K N A K,akik nem szeretnek a gepikoddal bajlodni.)
```

4. lista

```
FUNCTION:

         jsr&o473          ;charget
         jsr&96a5          ;adress
&o332:lda oo
         sta &14           ;x-mento
         lda 93
         jsr&ffd2          ;chrout
&o33b:jsr&ffcf             ;fn be
         cmp od            ;sor vege?
         beq &o34c
         ldx&14
         sta&cim,x         ;cim/a felhasznalo oda teszi,ahova akarja/
         inc &14
         jmp &o33b
&o34c:lda &3b
         pha
         lda &3c
         pha               ;mentes
         lda cim*
         sta &3b
         lda cim*
         sta &3c
         jsr &8953         ;tokenizalo rutin
         lda ff
         sta &81           ;jelzo
         jsr &9a9d         ;deffn rutin
         pla
         sta &3c
         pla
         sta &3b           ;mutatok vissza
         rts               ;vissza a basic programhoz
```

Listánkat felhasználói, illetve játékprogramokból állítjuk össze. A legjobbakat, legérdekesebbeket a beküldött javaslatok alapján rangsoroljuk. Ehhez kérjük az olvasók közreműködését. C64-re, ZX-Spectrumra, Enterprise-ra, TVC-re, Atari-ra és IBM-re készült programokat várunk havonta.

Címünk: Mikroszámítógép Magazin Szerkesztősége
1371 Budapest, Pf. 433

Diákszerkesztőség

Top lista

Felhasználói		Játék	
1. Geos 1.6 double	C64	1. Test Drive	C64
2. Print Fox	C64	2. Spy ver Spy 3.	C64
3. Giga Cad	C64	3. Elite	ZX
4. Art Studio 1.2c	ZX,C64	4. Nodes of yesad	C64
5. GRDOS 1.0	C128	5. Archon *	C64
6. Music Master	C64	6. Renegade	ZX
7. News room	C64	7. Suicide Express	Ent.
8. Rockmen v5.0	C64	8. The guild of time	C64
9. Letter design	C64	9. Slap Flight <P&A>	C64
10. Disk manager	C64	10. Defender etc. *	Amiga

EPSON®

Az EPSON RX—80-as típusú mátrixnyomtatót nem azért ajánljuk Önöknek, mert világbajnok volt a tartós nyomtatásban (2904 órát megszakítás nélkül nyomtatott), még csak azért sem, mert csupán három gomb is elegendő a kezeléséhez, vagy csak hozzá kell illeszteni személyi számítógépéhez, és egy megfelelő programmal máris üzemképes lesz, hanem azért:

— mert **megbízható,**
— mert **nagy teljesítményű** (100 karakter/s),
— mert **könnyű** (5 kilogramm),
— mert a **nyomtatófej** és **festékszalag-kazetta hosszú élettartamú,**
— mert egykártyás felépítésű és **könnyen javítható,**
— mert karakterkészlete tartalmazza a **magyar ábécét** is,
— mert **53 vezérlőparanccsal** rendelkezik,
— mert **96 ASCII karaktert, 32 grafikus karaktert és 11 nemzeti karaktert tartalmaz,**
— mert a cég a **nyomtatópiac 35,5 százalékát uralja,**
— mert **12 hónapig garanciális,**
— mert az EPSON nyomtató **más, mint a többi . . .**

Folyamatosan kapható valamennyi Centrum Áruház Műszaki Osztályán!

	Fogy. ár:
EPSON RX—80 mátrixnyomtató	56 400,— Ft
Ékezetes magyar ABC E—PROM BŐVÍTŐ	2 030,— Ft
Centronics PC kábelcsomag	4 730,— Ft
Enterprise—128 kábelcsomag	4 730,— Ft
Commodore soros interface kábel- és softwarecsomag	18 000,— Ft
Commodore 600/700 PIC printer interface chain, kábel- és softwarecsomag	20 800,— Ft
TV computer nyomtatókábelcsomag	1 630,— Ft

Figyelem! Valamennyi kábelcsomag mellé díjtalanul adjuk a Data Becker sorozat Az EPSON nyomtatók című könyvét.

KERESSE ÁRUHÁZAINKBAN AZ EPSON RX—80-as típusú mátrixnyomtató részletes műszaki tájékoztatóját!

BASIC

ÉS GÉPI KÓD

Legutóbb egy olyan gépi kódú rutint mutattam be, amely egy, az A/X regiszterpárba töltött, előjel nélküli egész számot BCD formájúra alakít, majd ASCII alakban kiírja a képernyőre. Most ennek a módosított változatát ismertetem, a hozzá tartozó BASIC betöltőprogrammal együtt.

A MÓDOSÍTOTT RUTIN

Az előző részben közölt rutinnak egyik hibája, hogy a számot elöl felesleges nullákkal feltöltve írja ki. Az *1. listán* a kiíratás módosított változatát láthatjuk, a megszokott disassemblált formában. Az átalakító rész (a $0370—039D címtartományban) változatlan, felesleges megismételni.

Az előző számban közölt kiíratás részben feltételeztük, hogy az X regiszter nulla kezdőértékkel érkezik. A jobb kezelhetőség érdekében eltekintünk ettől, és X-et a kiírás előtt nullázzuk. Ugyancsak nullát teszünk a $0365 címen levő munkaváltozóba.

Az ORA #$30/JSR $FFD2 utasításpárok helyére egy belső szubrutin hívását tesszük. Ez a $03BC címen kezdődő rutin oldja meg ravasz módon a vezető nullák elhagyását. Megvizsgálja, hogy az A regiszter tartalma megegyezik-e a $0365-ön lévő értékkel. Csak vezető nulla esetén van egyezés, ilyenkor a vezérlés írás nélkül adódik a hívást követő utasításra. Az első nullától eltérő érték észlelése után $0365-re minden további lépésben $30 és $39 közötti érték kerül, míg az A regiszterben az összehasonlításkor legfeljebb 9 lehet, tehát nem lehet egyezés, így minden további számjegy (0 is) kiíródik.

Az átalakító-kiíró rutin JSR $0370 utasítással hívható. Ugyanazt a feladatot látja el, mint az a ROM-rutin, amelynek hívási címe C64-en $BDCD, VC20-on $CDCD, C16-on pedig $A45F. Eltérés részint a végrehajtási sebességben van, részint abban, hogy a ROM-rutin megváltoztatja a FAC tartalmát, ezért nyomkövető rutinokhoz és hasonlókhoz nem használható.

A rutint Roger Harrisnak a *Compute!* 1984. júniusi számában megjelent *TRACE* programja alapján készítettem.

A BETÖLTŐPROGRAM

A *2. listán* látható az átalakító és a kiíró rutin BASIC betöltőprogramja, amely VC20-on, C64-en és C16-on egyaránt futtatható. Visszatértem a hagyományos decimális DATA-értékekhez. Ha valakinek megfelel a disassembly listákon meghatározott elhelyezés, annak elegendő a programot a 270-es sorig begépelni; erre utal a listán az üres sor. Aki máshová akarja a rutinokat tölteni, annak a teljes programot be kell írnia, és a 110-es sorban megadott értékeket is módosítani kell. KC-vel a program, AC-vel az adatterület kezdőcímét jelöltem. A program hossza 90 bájt, az adatok 6 bájtot foglalnak le. A kazettapuffer elejére történő elhelyezéshez KC=828, AC=918 szükséges.

Ha a rutinok részére a BASIC-területből vágunk le, a BASIC-mutatókat is módosítani kell az „ismert módon".

A működést BASIC-ből a POKE 780,HI:POKE 781,LO:SYS KC utasítássorozattal lehet kipróbálni, ahol az átalakítandó szám magasabb helyértékű bájtjának értékét a HI, az alacsonyabbét a LO, a program kezdőcímét a KC jelöli. A kipróbálástól eltekintve a BASIC-ből való hívásnak nincs értelme, helyette a PRINT utasítást kell használni. Ha csak az átalakító rutint akarjuk használni, a 230-as sorban a 234-es számot cseréljük 96-ra, a sor végén álló 1569-es ellenőrző számot 1431-re.

```
039e ea           nop
039f a2 00        ldx #$00
03a1 8e 65 03     stx $0365
03a4 bd 60 03     lda $0360,x
03a7 4a           lsr
03a8 4a           lsr
03a9 4a           lsr
03aa 4a           lsr
03ab 20 bc 03     jsr $03bc
03ae bd 60 03     lda $0360,x
03b1 29 0f        and #$0f
03b3 20 bc 03     jsr $03bc
03b6 e8           inx
03b7 e0 03        cpx #$03
03b9 d0 e9        bne $03a4
03bb 60           rts
03bc cd 65 03     cmp $0365
03bf d0 01        bne $03c2
03c1 60           rts
03c2 09 30        ora #$30
03c4 8d 65 03     sta $0365
03c7 4c d2 ff     jmp $ffd2
```

1. lista

VÉGÜL

Befejezem — legalábbis egyelőre — a decimális aritmetika ismertetését. A további munkához szeretnék néhány ötletet adni azoknak, akiknek sikerült felkeltenem az érdeklődését a téma iránt.

Mint említettem, a BCD-aritmetikát elsorban gépi kódú programokban érdemes használni. BASIC-ből a nagy pontosságú, úgynevezett *sztringaritmetika* alkalmazása célszerű. Ez azt jelenti, hogy karakterláncokban lévő decimális számokkal végzünk aritmetikai műveleteket, természetesen megfelelő átalakítások után. Érdekes feladat a szorzás és osztás BCD-változatának kidolgozása is.

Példánkban a BCD-számokat természetes sorrendben tároltuk, vagyis a magasabb helyértékű jegyek álltak elöl. Tapasztalataim azt mutatják, hogy a fordított sorrend technikailag egyszerűbb megoldásokhoz vezet. Remélem, nem okoz problémát, hogy a Commodore BASIC az egész számokat természetes sorrendben tárolja, nem cimformátumban.

A BCD-aritmetika további alkalmazási területe a játékprogramok, ahol pontszámlálásra használható. Ehhez az eddig leírtak elegendő információt nyújtanak.

BARNA LÁSZLÓ

```
100 rem        BCD rutinok
110 kc=880 : ac=864
120 z=256 : print chr$(147)
130 for i=kc to kc+80 step 10 : s=0
140 for j=0 to 9
150 read a : poke i+j,a : s=s+a : next
160 print chr$(19);peek(63)+z*peek(64);
170 read b : if b<>s then print "hiba!" : stop
180 next
190 data 142,99,3,141,100,3,169,0,141,96,894
200 data 3,141,97,3,141,98,3,162,16,14,678
210 data 99,3,46,100,3,160,2,185,96,3,697
220 data 120,248,121,96,3,216,88,153,96,3,1144
230 data 136,16,240,202,208,229,234,162,0,142,1569
240 data 101,3,189,96,3,74,74,74,74,32,720
250 data 188,3,189,96,3,41,15,32,188,3,758
260 data 232,224,3,208,233,96,205,101,3,208,1513
270 data 1,96,9,48,141,101,3,76,210,255,940

280 for i=1 to 17 : c=ac : if i>15 then c=kc
290 read d,a : a=c+a : b=int(a/z)
300 poke kc+d,a-z*b : poke kc+d+1,b : next
310 data 1,3,4,4,9,0,12,1,15,2
320 data 20,3,23,4,28,0,33,0,38,0
330 data 50,5,53,0,63,0,77,5,85,5
340 data 60,76,68,76
```

2. lista

Néhány nyelvről – néhány szóban

	Megvalósítás	Olvashatóság	Bővíthetőség	Strukturáltság	Fejlesztői környezet	Elsajátíthatóság
FORTRAN	Ford.	Gyenge	Közepes	Gyenge	Közepes	Közepes
COBOL	Ford.	Közepes	Gyenge	Gyenge	Jó	Jó
BASIC	ért.	Gyenge	Rossz	Rossz	Jó	Jó
PL/I	Ford.	Jó	Közepes	Közepes	Gyenge	Gyenge
LISP	ért.	Rossz	Jó	Jó	Jó	Rossz
Pascal	Ford.	Jó	Jó	Jó	Jó	Közepes
C	Ford.	Jó	Jó	Jó	Jó	Közepes
FORTH	Vegyes	Gyenge	Jó	Jó	Jó	Közepes
LOGO	ért.	Közepes	Jó	Jó	Jó	Jó
PROLOG	ért.	Jó	Jó	Jó	Jó	Közepes

A Fordítás vagy értelmezés c. cikkem (lásd a Magazin 1988/7. számában) befejezésekor döbbentem rá, hogy ebben a rövid írásban tíz magas szintűnek nevezhető programozási nyelvet említek, melyek ma is – ha nem is azonos mértékben – használatban vannak.

Magazinunk hasábjain, főleg a „Mit tud a . . ." sorozatban már találkozhattunk néhány nyelvvel, most mégis úgy érzem, hogy a jelenleg legnépszerűbb nyelvek egy rövid – bizonyos mértékig összehasonlító – jellemzésére is szükség van.

A nagyobb teljesítményű, professzionális személyi számítógépek hazai térhódítása abba a kellemes helyzetbe hozza az érdeklődőket, hogy programjaikat már nemcsak BASIC-ben (esetleg assemblyben) írhatják, hanem egy tucat nyelv között válogathatnak.

Ha egy programozási nyelvet röviden akarunk bemutatni, az elméleti alapokkal és a jellemző alkalmazási területekkel célszerű kezdeni. Az összevetés érdekében pedig néhány szempont szerint táblázatot állítottunk össze. A tárgyalt nyelvek két jellemzőjükben megegyeznek:

— mindegyikük magas szintű nyelv,

— minden IBM PC-szerű gépen rendelkezésre állnak, némelyikük a nyolcbites gépeken is használható.

A tárgyalás sorrendje nagyjából a nyelvek megjelenéséhez igazodik, nincs kapcsolatban a nyelv népszerűségével.

FORTRAN

Az ötvenes évek közepén született, amikor a számítógépeket még főleg számolásra használták. Erre utal neve is (FORmula-TRANslation = formula-fordítás). Eredeti formájában elsősorban numerikus algoritmusok programozására volt alkalmas nagygépes környezetben, de az elmúlt harminc év alatt sokat változott a modernebb alkalmazások, a személyi számítógépek és a strukturált programozás irányába. Ennek legjobb bizonyítéka, hogy ma is él. Fennmaradását elsősorban viszonylag egyszerű szerkezetének, a nagy számítógépgyártók támogatásának és annak a ténynek köszönheti, hogy az évtizedek során igen jelentős „szoftvervagyon" halmozódott fel FORTRAN-ban. Ezen a nyelven készült például egy, már mintegy huszonöt éve folyamatosan fejlesztett statisztikai programcsomag, amely ma is egyedülálló a világon.

LISP

Az ötvenes évek végén egyes amerikai egyetemeken a mesterséges intelligencia kutatása már ott tartott, hogy reálissá vált egy saját programozási nyelv létrehozása is. A témával elsősorban matematikusok foglalkoztak. A nyelv alapelvei J. McCarthy professzor szerint a következőkben foglalhatók össze:

— számokon végzett műveletek helyett szimbolikus kifejezések (formulák) kezelése,

— a formuláknak és minden más információnak a gép memóriájában listaszerkezettel való ábrázolása (ez általában többszintű listákat jelent),

— feltételes kifejezések és függvények (gyakran rekurzív) használata,

— a logikai műveletek teljes körű alkalmazhatósága,

— egy speciális kiértékelő függvény (eval) bevezetése, amely egyaránt szolgál a nyelv definíciójára és értelmezésére,

— speciális memóriagazdálkodási stratégia a feleslegessé vált listaelemek által elfoglalt tároló újbóli felhasználására (ezt röviden „szemétgyűjtésnek" nevezik).

A LISP tehát a hagyományos elvektől sokban eltérő, a matematikai logika eszköztárára épülő nyelv, amely speciális alkalmazási területe ellenére is több módon hozzájárult a programozási nyelvek fejlődéséhez. A mesterséges intelligenciával kapcsolatos alkalmazásokban (például szakértői rendszerek) ma is igen népszerű.

COBOL

Amerikai üzletemberek kezdeményezésére jött létre 1960/61-ben nagygépes környezetben folyó ügyviteli tevékenységek programozásának támogatására. A nyelv neve (COmmon Business Oriented Language = közös ügyviteli nyelv) és szerkezete is az ügyvitel-gépesítést sugallja. A programok nagy részét a beolvasásra, illetve kiírásra kerülő adatok szerkezetének az eredeti kérdőívekhez igazodó leírása alkotja. A programok tényleges tevékenysége a FORTRAN-nal ellentétben főleg adatmozgatásra irányul, csak a legszükségesebb aritmetikai lehetőségeket tartalmazza. A nyelv fejlesztése során először az adatbázisokra alapozott adatkezelést valósították meg, majd a mikrogépes környezetben is kiharcolták a COBOL helyét azzal, hogy interaktív, teljes körű képernyőkezelést megengedő fejlesztői környezetet hoztak létre. E fejlődés talán legékesebb bizonyítéka a nyelv PROFESSIONAL COBOL nevű leszármazottja.

BASIC

Nagyjából a COBOL-lal egykorú, de nem ipari, hanem egyetemi környezetben született azzal a céllal, hogy kezdők számára megkönnyítse a számítástechnikával való ismerkedést. A nyelv egyik létrehozója J. Kemény, magyar származású professzor volt. Mivel a nyelvet eredetileg csak jobbára kezdő hallgatók oktatására használták, három követelményt igyekeztek elsősorban kielégíteni:

— legyen egyszerű felépítésű, gyorsan elsajátítható és számítógépeken könnyen megvalósítható,

— legyen univerzális: numerikus és szöveges információ feldolgozására egyaránt alkalmas,

— a nyelvhez gyors eredményt adó, inkrementális programfejlesztést lehetővé tevő értelmező rendszer tartozzon, amely minden bevitt programsort azonnal ellenőriz és kijavíttat.

A BASIC évekig nem volt és nem is akart több lenni oktatási segédeszköznél. Amikor a mikroprocesszorok kora elkezdődött, a laikus felhasználók tömegei éppen a fenti három tulajdonsága miatt vették boldogan birtokukba és használják azóta is.

PL/I

Az IBM a 360-as gépcsalád létrehozásakor, a hatvanas évek közepe táján úgy érezhette, hogy az új hardvergeneráció egy új nyelvet is érdemel. Az új nyelvben egyesíteni akarták az akkor népszerű ALGOL, COBOL és FORTRAN lehetőségeit feltehetően azzal a céllal, hogy tudományos, műszaki és ügyviteli feladatokra egyetlen univerzális nyelvet adjanak ügyfeleik kezébe. Ma már sokan tudják, hogy ilyen — minden célt egyformán elérő — nyelvet valószínűleg nem is lehet, de semmiképpen nem érdemes erőltetni. Ugyanakkor az IBM kutató-fejlesztő műhelyeinek dicséretére vált a feladat megoldásának magas szakmai színvonala. Mintegy „mellékterméként" például a bécsi laboratóriumban világhírű eredményeket értek el formális nyelvek leírására szolgáló eszközök kidolgozásával kapcsolatban. A jó elméleti alapok mellett a PL/I terjedését elsősorban a „Nagy Kék" gazdasági súlya, hatalmas piaci részesedése és tekintélye segítette elő. A nyelvet (természetesen egy szűkített változatát) a személyi számítógépeken is használhatjuk, bár úgy tűnik, népszerűsége csökkenőben van.

PASCAL

Az azóta már majdnem elfelejtett ALGOL első krízise után, 1970 táján N. Wirth professzor fektette le a Pascal alapjait azzal a szándékkal, hogy mind az egyetemi oktatás, mind pedig az általános alkalmazás számára praktikus nyelvet hozzon létre. Ennek érdekében a tudományos megalapozást csak olyan mértékben erőltette, hogy az ne menjen az érthetőség és a viszonylag egyszerű gyakorlati alkalmazhatóság rovására. Eredményül mikrogépeken is jól megvalósítható, a korszerű programozási elveket követő, gyorsan népszerűvé vált nyelvet kapott. A Pascal valóban univerzális nyelvnek nevezhető, amely egyrészt újat hozott a programozásba, másrészt az akkori, különböző nyelvekben meglévő „elszigetelt" értékeket igyekezett egy csokorba kötni.

Talán legfontosabb „újítása" a szabad felhasználói típusdefiníció. Ez lehetőséget ad arra, hogy néhány egyszerű alaptípusból kiindulva a programozó fokozatosan építkezve létrehozhassa (és elnevezhesse) azt az adattípust, amely az adott feladathoz a legjobban illeszkedik. A korszerű eszközök beépítésének jó példái a mutatók, a strukturált utasítások és a rugalmas eljárásfüggvény fogalom.

A Pascal népszerűsége talán még ma is növekszik, amit az is bizonyít, hogy számos változata van forgalomban.

C

Megint egy amerikai „származású" nyelvről van szó, amelynek bölcsőjét az AT&T kutatóbázisán, a Bell Laboratóriumban ringatta néhány lelkes szoftveres. Ez a tény tulajdonképpen meghatározza a C nyelv legfontosabb jellemzőjét is: rendszerprogramozók készítették rendszerprogramozók számára, azaz teljes hatékonysággal csak a gépközeli programozásban járatosak tudják alkalmazni.

Mint rendszerprogramozási nyelv, a C egyesíti a magas szintű programozás kényelmét a gép minden porcikájához való közvetlen hozzáférés előnyeivel. Ezt leginkább az igen általános és jól használható címaritmetikája és a mutatók (indirekt címzés) segítségével éri el. Rendelkezik minden fontosabb strukturált utasítással, a programot egy kitüntetett és ebből hívható további függvények halmazaként kell felépíteni. A nyelv gépi megvalósításai általában kétlépcsős fordítást végeznek, ahol először egy assembly programot kapunk, amit — legtöbbször automatikusan — a gép assemblere fordít tovább.

FORTH

A rendszerprogramozás egy igen érdekes megközelítését képviseli ez a nyelv, amelynek talán legnagyobb erénye a hatékonyság. Kidolgozója — amint ezt már megszokhattuk — egy amerikai programozó, C. H. Moore. A FORTH kialakulása mintegy tíz évig tartott, az első teljesnek nevezhető változat 1973-ra készült el, amikor is a szerző egy vállalatot alapított a nyelv lehetőségeinek kiaknázására. A FORTH a programfejlesztés hatékonyságát elsősorban néhány jó meglátásnak köszönheti:

— A gépi megvalósítás, természetesen FORTH nyelven, a fordítás és értelmezés egy-egy jó keverékét alkalmazza. A fordító a forrásprogram betöltése közben egy igen tömör, könnyen értelmezhető belső kódot állít elő, melyet egy igen kisméretű és gyors értelmező hajt végre.

— A program tulajdonképpen független, de egymással kommunikáló, láncba fűzhető eljárások (FORTH terminológiával: szavak) halmaza, melyek tetszőleges mélységben skatulyázhatók. A rendszer kezdetben csak az alapszavakat (mintegy 140 darab) tartalmazza, egy adott alkalmazáshoz a programozó a szókincset a feladatnak megfelelően bővítheti. Ily módon egy-egy alkalmazás csak a ténylegesen szükséges eszközöket tartalmazza.

— Az eljárások közötti információcsere veremmemóriák segítségével igen egyszerűen és hatékonyan valósul meg.

Bár a FORTH eredetileg rendszerprogramozók számára készült, népszerűsége folytán univerzális nyelvvé fejlődött. Kibővítették háttértár-kezeléssel, lebegőpontos aritmetikával, grafikával és egyéb általános programozási eszközökkel is.

LOGO

Egy amerikai egyetem munkatársai dolgozták ki a nyelvet, elsősorban alapfokú számítástechnikai ismeretek oktatására, de nemcsak egyetemi hallgatók, hanem a legfiatalabb korosztály számára is. Ennek megfelelően a LOGO igen barátságos arcot mutat a felhasználók felé, de ugyanakkor tartalmazza a korszerű számítástudomány minden fontosabb eszközét. A nyelv négy alappillére egy jól kidolgozott aritmetika, a teljesen általános és rekurzív módon is definiálható eljárások, a listakezelés és egy igen szellemes, az irodalomban „teknőcgrafika" néven elterjedt eszköztár, amely a fiatalok érdeklődésének felkeltésére különösen alkalmas.

A LOGO az alapeljárásaival, amelyeket „primitiveknek" nevez, a FORTH-ra emlékeztet. A primitívek halmaza azonban lényegesen bővebb és más jellegű, a megvalósításnál pedig a didaktikai szempontokat helyezték a hatékonysági megfontolások elé. Az eredmény egy modellezésre, tanulásra és gyakorlásra igen jó, de nem ipari jellegű rendszer.

PROLOG

Meglepő lenne, ha a mesterséges intelligencia művelői egy népszerű nyelvvel megelégednének. Ez a tíz közül a legfiatalabbnak tekinthető nyelv több szempontból újszerű. Kialakulása 1973-ra tehető, és meglepő módon a bölcső nem Amerikában, hanem Angliában és Franciaországban ringott. A PROLOG logikai alapú nyelv, amely új elvűnek is nevezhető abban az értelemben, hogy a programozó csak a megoldandó feladatot írja le, a megoldás menetét (algoritmusát) nem kell megadnia. A feladatot természetesen a matematikai logika eszközeivel olyan formában kell megfogalmazni, amely kimondja, hogy létezik egy, a kívánt feltételeknek eleget tevő fogalom. Magát a fogalom konstrukcióját a matematikai logika formuláinak egy alkalmas részhalmazán működő tételbizonyító, a PROLOG értelmező végzi. Maga a program a feladathoz tartozó axiómarendszernek felel meg, az utasításokat eldöntendő állítások reprezentálják.

A mesterséges intelligencia gyakorlati alkalmazása során a LISP-nél barátságosabb PROLOG-nak nagy szerepet szántak: 1981-ben a japán ötödik generációs számítógéprendszer tervezetében ezt jelölték meg az új számítógépcsalád alapnyelvéül. Ennek köszönhetően a nyelv népszerűsége nagymértékben növekedett.

BAKOS TAMÁS

A programozás alapkérdései III.

Ötrészes sorozatunkban mindarról szót ejtünk, amit olvasóinknak érdemes átgondolniuk, ha programozásra adják a fejüket, vagy ha valamely programról véleményt kívánnak alkotni. Ezek az írások a szerzőnek lapunkban közölt korábbi fejtegetéseivel együtt (utalunk „Az adattípus fejlődése" és a „Függvények és utasítások" című sorozatokra, lásd 1987/8. és 9., valamint 1987/12., 1988/1. és 2. számainkat), teljeskörűen és közérthetően elemzik e sorozat címében jelzett (problematikát) témát.

EGY FÜGGVÉNY – EGY CÉL

A program kisebb-nagyobb egységekből áll: modulokból, függvényekből, utasításokból. Az utasítások értelmét a nyelv szabja meg, és a programozó arra használja őket, amire akarja. A legkisebb olyan egység, amelynek a célját a felhasználó szabja meg, a függvény vagy eljárás. Egy függvény tudhat nagyon sokat és nagyon keveset is: például megadhatja egy teljes egyenletrendszer megoldását, de van olyan is, melynek funkciója mindössze egy szám négyzetre emelése.

A függvényeknek és eljárásoknak a felhasználó nevet ad. A névadás fontos dolog, ez választja el a magas szintű nyelvek eljárásait az alacsonyabb szintű nyelvek, például a BASIC és a gépi kód szubrutinjaitól.

SZUBRUTINOK ÉS FÜGGVÉNYEK

A szubrutin olyan utasássorozat, amelyet úgy használunk, hogy „ráugrunk", megjegyezzük az elugrás helyét, és amikor befejeződött a szubrutin végrehajtása, visszatérünk oda.

Szubrutint olyan részfeladatokra dolgozunk ki, amelyek sokszor megismétlődnek a programban, és a lépéseket lusták vagyunk mindig leírni — továbbá így rövidíthetjük is a program szövegét. Nagyon egyszerű szubrutinok minden előkészítés nélkül meghívhatók, és a végrehajtás után semmi munkánk sincs velük. A főprogram rendszerint jól meghatározott helyeken előkészít adatokat a szubrutinnak, és a munka végén átveszi az eredményeket. A főprogramnak nemcsak azt kell tudnia, mit csinál a szubrutin, hanem azt is, hogyan akarja: milyen adatot és honnan vesz, mit használ fel. Ugyanígy a szubrutinnak is tudnia kell, mit hol tart a program. Nézzünk például egy BASIC szubrutint, ami egy grafikus képbe feliratokat helyez el. Előtte szükségszerűen két változóba elhelyezendők a kezdőpont koordinátái, és egy sztringbe kell tenni, hogy mi legyen a felirat. Futás után a két koordinátaváltozóban megkapom a szöveg végének koordinátáit. Ez a szubrutin majdnem olyan szép, mint egy függvény: vannak függvények, amelyek ennél sokkal jobban összefonódnak a főprogrammal vagy egymással.

Egy helyesen megírt függvény vagy eljárás egészen más valami. Egy függvény olyan, önmagában zárt és jól definiálható valami, ami akármelyik programba betehető, és ott úgy viselkedik, ahogy a leírása mondja, anélkül, hogy a felhasználó követhetné, milyen utasításokat hajt végre. A jó eljárás csak az IN paraméterein keresztül vesz adatokat a külvilágtól és csak az OUT paramétereken keresztül ad vissza. A munkájához szükséges lokális munkaváltozókat maga hozza létre, ha kell, és maga szünteti meg a munka végén.

Egy jó függvény tehát elsősorban nem a program gyakran használt utasássorozata, hanem lényegében absztrakció. Olyan tevékenység, amelyről önmagában beszélni lehet, nevet lehet neki adni, ki lehet venni a programból, el lehet tenni egy könyvtárba, és onnan máskor ismét felhasználható.

AZ ELNEVEZETT PROGRAMRÉSZEK

Egy modern programban sok olyan függvény és eljárás van, amelyeket csak egyszer hívnak meg a programban. Ez így, első hallásra elég bizarr dolognak tűnhet, hiszen a függvényhívás nem olcsó mulatság: egy felhasználó által írt függvény meghívása és a belőle való visszatérés általában 30-60 közönséges utasítás végrehajtásával egyenlő. Egy függvényhívás a következő lépésekből áll: elugrás (miközben megjegyezzük a helyet, hogy honnan), a lokális adatterület létrehozása, a paraméterek átvitele. A függvény végrehajtása után az eredmény visszaadása, a lokális adatok felszámolása, visszaugrás, végül — az eredmény felhasználása után — az eredmény megszüntetése.

Sajnálatos módon a modern programozási nyelvekben két dolog összekeveredik a függvény fogalmában: egyrészt az, hogy a program egy részének nevet adjunk és elhatároljuk a program többi részétől, másrészt pedig, hogy bizonyos, gyakran ismétlődő, hasonló tevékenységeket megfogalmazzunk egy megfelelően általános alakban, és egy bármikor felhasználható új eszközt — egy függvényt — hozzunk létre belőle. (Egyetlen olyan nyelvet ismerek, az ELAN nyelvet, amely megkülönbözteti a két absztrakciót. Ezt a nyelvet azonban csak egyetemeken használják, kezdő programozók oktatására.)

MI KERÜLJÖN EGY FÜGGVÉNYBE?

Valamely programban nagyon ritkán fordul elő, hogy két különböző helyen pontosan ugyanazt az algoritmust kelljen végrehajtani, apróbb különbségek mindig adódnak. Ezért ahhoz, hogy a függvényt megfogalmazzuk, bizonyos absztrakcióra van szükség, és a felhasználó dönti el, hogy két hasonló tevékenységből egy függvényt csinál-e vagy sem. Például: vektorokkal kell számolnunk és két gyakran előforduló teendőnk van: meg kell állapítani, hogy két

vektor merőleges-e, illetve meg kell határoznunk egy vektor hosszát.

A feladatot két különböző függvénnyel is megadhatjuk, és eggyel is. A merőlegességet annak alapján szoktuk megállapitani, hogy két vektor skalárszorzata nulla, a vektor hosszát pedig az önmagával vett skalárszorzatból vont négyzetgyök adja. A problémát természetesen úgy is meg lehet oldani, hogy először bevezetjük a skalárszorzatot, azután ennek segitségével egy igaz vagy hamis értéket visszaadó, merőlegességvizsgáló függvényt, továbbá egy hosszat megállapító függvényt, amely egyetlen gyökvonásból áll. A megoldás roppant elegáns, de hatékonysága csapnivaló, és teljesen felesleges absztrakciókat vezet be. Persze az absztrakciónak nehéz határt szabni . . .

Sok programozó felfedezte már, hogy egy listán vagy másféle halmazban a minimum és a maximum megkeresése pusztán egy feltétel megfordításán múlik. Írtak is olyan függvényt, amely egy paraméter beállításától függően vagy minimumot, vagy maximumot keres. Ez tulajdonképpen még el sem itélhető igazán, ha elég olvasható a függvényhivás — például az a neve, hogy EXTRÉMUM —, és a keresést irányító paraméter csak a MIN vagy MAX szócska lehet. Azt már senkinek sem ajánlanám azonban, hogy a függvény a minimum és maximum mellett még az átlagot is megállapitsa.

A gond rendszerint akkor kezdődik, amikor egy bizonyos függvényt eléggé össze nem tartozó dolgokra akarunk felhasználni. Például, ha karaktersorozatokat manipulálunk, két feladatunk van: egy karaktertömböt szóközzel kitölteni és egy karaktersorozatot átmásolni valahova. Ennek szokásos trükkös megoldása a következő. Írunk egy eljárást, ami átrakja az elejétől kezdve a karaktereket az egyik tömbből a másikba. Ha a tömböt szóközökkel akarjuk feltölteni, az első elemébe egy szóközt teszünk, és átmásoljuk a tömbnek az elsőtől az utolsó előtti eleméig tartó részét a másodiktól az utolsó karakterig tartó részébe. Ekkor a szóköz átmásolódik az első elemből a másodikba, a másodikból a harmadikba és igy tovább.

Miért rossz ez a megoldás? Mert baj van, amikor a programon valamit változtatni kell. Jön egy kivülálló, aki nem tudja, milyen trükköket alkalmaztunk; nézi a trükköt, nem érti, ki tudja, meddig gondolkozik, amig rájön, hogy az átmásoló függvény belsejét is meg kell néznie ahhoz, hogy megértse, mi történik. A másik eset még rosszabb. A javitást végző utód látja, hogy van egy másolófüggvény, és ő egy karakterláncba be akar szúrni valamit. Úgy gondolja, hogy ez a következőképpen tehető meg: a karakterlánc végét átmásoltatja másolóval a megfelelő számú karakterrel hátrébbra, és a lyukba beírja a beszúrandó részt. Meglepve tapasztalja, hogy valami nem várt zagyvaságot kapott.

A problémát itt tehát az okozta, hogy nagyon leragadtunk az adott feladatnál, nem vittük végig az absztrakciót — azaz, hogy mit kell tudnia egy általános sztringmásoló függvénynek —, amivel megtévesztettük a későbbi alkalmazókat.

Hasonlóan egészségtelen megoldás, ha egy függvény egyszerre több mindent képes csinálni. Nem szerencsés például olyan eljárást irni, ami ha kell, letörli a képernyőt, rááll a képernyő egy adott pontjára, majd ott valamilyen betütipussal (inverz, villogó, aláhúzott stb.) kiir valamilyen szöveget. Az a tapasztalat, hogy ezeket a csodadolgokat az alkalmazók „túlhasználják", ezeket hivják meg a képernyő letörlésére, a képernyőn való pozicionálásra, arra, hogy a képernyőre különleges betütipussal irjanak. Később annak, aki valamilyen nagyobb rendszeren javitani akar, megőszülhet minden haja szála, amig rájön, hogy miért hivták meg ezt a bizonyos függvényt, azokkal a látszólag értelmetlen paraméterekkel.

Igaz, hogy három függvényhivás kevésbé hatékony, mint egy, viszont az esetek jó részében úgy sincs mindegyikre szükség. Mindig arra kell gondolnunk, hogy a programok nemcsak a mának, hanem a holnapnak is készülnek. Az olvashatóság és javithatóság majdnem olyan fontos, mint a hatékonyság.

A fentiekből talán érzékelhetővé vált, hogy egy függvény akkor hasznos igazán, ha már a nevéből rá lehet jönni, hogy mi célt szolgál. Ez persze nem mindig lehetséges, de a legtöbbször egyetlen összetett mondattal elmondható, mit csinál az a bizonyos függvény. Ezután még néhány mondat szükséges ahhoz, hogy pontosabbá tegyük a paraméterátadás módját, és hogy a működéskor milyen hibák léphetnek fel. Ha egy függvény meghivása túl komplikált, akkor azt még a saját irója is elronthatja majd egyszer-kétszer.

KONKLÚZIÓ

Ebben a részben azzal a kérdéssel foglalkoztam, hogy miből érdemes függvényt létrehozni egy programban, és miből nem. Az előrelátó programozó nem arra törekszik, hogy minél kényelmesebben dolgozzon, hanem igyekszik — talán egy kicsit több munkával — a feladat megoldásához olyan eszközkészletet létrehozni, ami nemcsak az adott feladathoz jó, hanem később is használható.

HOMÁLYOS ÜGYEK

Amit a korábbiakban mondtunk, természetesen elsősorban a valódi függvényekre érvényes, és sokkal kevésbé érvényes azokra a függvényként megadott, elnevezett programrészekre, amelyekről előzőleg beszéltünk. Szinte minden program inicializálással kezdődik, ahol letöröljük a képernyőt, megnyitjuk a beolvasandó és kiirásra váró állományokat, kezdőértéket adunk bizonyos változóknak — ezek bizony nem szorosan összetartozó dolgok. Az elnevezett programrészek általában nem annyira absztrakt dolgok, nem függetleníthetők annyira a konkrét programtól, mint az igazán jól kiválasztott függvények és eljárások.

A másik homályos ügy a hibajelzésekkel függ össze. Mit csináljon a függvény, ha valamilyen okból nem sikerült jól lefutnia? Maga adjon-e hibajelzést a felhasználónak, vagy a hivó programnak jelezze vissza, hogy nem futott le jól?! Része-e a hibajelzés a hivott program céljának? Nekem az a tapasztalatom, hogy szélesebb körben használhatók azok a függvények, amelyek a főprogramnak súgnak, mint amelyek maguk küldenek hibajelzést. Kivéve, ha a függvény arra is önmaga nyújt lehetőséget, hogy a hibát azonnal kijavitsuk, és mindig helyes eredménnyel tér vissza a főprogramba.

A FÜGGVÉNYEK MÉRETE

Mint arról korábban szó volt, a függvényhivás nem olcsó mulatság. Kérdés, hogy mekkora az az utasitássorozat, aminek a kedvéért érdemes egy függvényt bevezetni. Ezt elég nehéz megmondani: egyként szekvenciális utasitásért biztos nem, de egy — még viszonylag rövid — utasitássorozatért, amiben ciklus van, lehet, hogy igen. Pszichológiai vizsgálatok szerint a programozók jól megértenek valamit, megjegyeznek viszonylag hosszú, egymást követő és elágazásmentes sorozatokat, viszont nem szeretik a többszörösen egymásba ágyazott (vezérlési) struktúrákat.

Akkor érdemes tehát függvényt bevezetni, ha annak a résznek az olvashatósága, ahonnan kivettük, javul, és a végrehajtás hatékonysága nem nagyon romlik. Ha a kivett részben van egy hosszú futásidejű ciklus, akkor a függvényhivás és visszatérés okozta többlet aránya nem lesz rossz. Nem tanácsos viszont egy ciklus magjából függvényt csinálni, mert ez lényegesen ronthatja a futásidőt.

MAKRÓK ÉS BESZÚRT FÜGGVÉNYEK

A programozót persze bosszantja, ha bizonyos három-négy-öt-hat utasitásból álló sorozatot sokszor kell leírnia. Ezen az sem segit sokat, ha jó a szövegszerkesztőnk, amivel könnyű megismételni ezeket az utasitássorozatokat, mert az olvasót is boszszantani fogja. Ennek elkerülésére szolgálnak a makrók vagy a beszúrt (inline) függvényhivások. Ezeket úgy kell elképzelni, hogy helyükre nem függvényhivás fordul a programba, hanem maga a végrehajtandó utasitássorozat kerül oda.

A makróknál a használni kivánt utasitások helyett egy rövidités vezetünk be, és ezt használjuk a program szövegében. Lefordítás előtt a makróhivás helyére bemásolódik a megfelelő utasitások szövege, és kifejtett szöveg fordítódik le.

Az inline függvényeknél a függvényt úgy fordítja le a fordítóprogram, hogy a függvényhivások helyére nem a szokásos függvényhivó utasitássorozat kerül, hanem a függvényt alkotó utasitások. Sajnos kevés olyan nyelv van, amelyik ilyén kényelmet nyújt. A kurrens nyelvek közül a C nyelvben van makró lehetőség, és az Ada nyelvben használhatunk inline függvényeket.

FARKAS ERNŐ

A nagyfelbontású képernyő forgatása bájtonként I.

1. lista

```
                        :
115:    CC00            AKT     =   $FB
120:    CC00            KOV     =   $FD
                        :
130:    CC00                    *=  $CC00
                        :
145:    CC00 A5 01      FORGAS  LDA #1
150:    CC02 8D BC 02           STA TEMP+2
155:    CC05 A9 34              LDA #$34
160:    CC07 78                 SEI
165:    CC08 85 01              STA #1
175:    CC0A AD B5 02           LDA KEP
180:    CC0D 8D BA 02           STA TEMP
185:    CC10 AD B6 02           LDA KEP+1
190:    CC13 8D BB 02           STA TEMP+1
200:    CC16 AD B8 02           LDA OSZLOP
205:    CC19 8D B9 02           STA OSZAM
215:    CC1C AE B7 02 UJOSZLP   LDX SOR
220:    CC1F A0 00              LDY #0
230:    CC21 AD BA 02           LDA TEMP
235:    CC24 85 FD              STA KOV
240:    CC26 AD BB 02           LDA TEMP+1
245:    CC29 85 FE              STA KOV+1
255:    CC2B B1 FD              LDA (KOV),Y
260:    CC2D 48                 PHA
270:    CC2E 18         FORGLOP CLC
275:    CC2F A5 FD              LDA KOV
280:    CC31 85 FB              STA AKT
285:    CC33 69 01              ADC #1
290:    CC35 85 FD              STA KOV
295:    CC37 A5 FE              LDA KOV+1
300:    CC39 85 FC              STA AKT+1
305:    CC3B 69 00              ADC #0
310:    CC3D 85 FE              STA KOV+1
315:    CC3F A0 00              LDY #0
325:    CC41 B1 FD      KARFORG LDA (KOV),Y
330:    CC43 91 FB              STA (AKT),Y
335:    CC45 C8                 INY
340:    CC46 C0 07              CPY #7
345:    CC48 D0 F7              BNE KARFORG
355:    CC4A 18                 CLC
360:    CC4B A5 FB              LDA AKT
365:    CC4D 69 40              ADC #64
370:    CC4F 85 FD              STA KOV
375:    CC51 A5 FC              LDA AKT+1
380:    CC53 69 01              ADC #1
385:    CC55 85 FE              STA KOV+1
395:    CC57 A0 00              LDY #0
400:    CC59 B1 FD              LDA (KOV),Y
405:    CC5B A0 07              LDY #7
410:    CC5D 91 FB              STA (AKT),Y
```

Az előző részben, a 88/9-es számban a nagyfelbontású képernyő bitenkénti forgatását ismertettük. Most a képernyő bájtonkénti forgatásáról lesz szó. Emlékeztetőül: a nagyfelbontású képernyő 320×200 képpontból (más néven pixelből) áll. Egy képpont egy bitnek felel meg. Így tehát vízszintesen az egy képpont szélességű, úgynevezett grafikus képsor megjelenítéséhez 320 bitre, azaz 40 bájtra van szükség. Függőleges irányban az egy pixel vastagságú, 200 képpontból álló pontsor megjelenítéséhez viszont már 200 bájtra van szükség. Ez azonban csak az első pillanatban meglepő, teljesen egyértelművé válik akkor, ha belegondolunk abba, hogy a képernyő bájtjai nem függőlegesen lógnak, hanem vízszintesen helyezkednek el. Természetesen ugyanezen a 200 bájton rögtön 8 függőleges képpontoszlopot is kigyújthattunk volna. Ehhez hasonlóan, ha csak egyetlenegy képpontot kívánunk kigyújtani a képernyőn, ennek az egy pixelnek a kigyújtásához is egy bájtra van szükség.

Ahogy azt az előző cikkben már jeleztük, a nagyfelbontású képernyő forgatása azért bonyolultabb például a sprite-ok forgatásánál, mert az ilyen képernyő bájtjainak számozása — a karakterek könnyebb megjeleníthetősége miatt — nem sorfolytonos, hanem a számozásban vízszintes irányban 8 bájtos bakugrások vannak. Függőlegesen pedig még faramucibb a helyzet, mert 8 bájton keresztül (ez a nyolc bájt, azaz 64 pixel alkot egy karaktert) a számozás folyamatos, majd a következő karaktersorba áttérve, a számozásban egy igazi nagy ugrás kö-

```
420:    CC5F CA                 DEX
425:    CC60 10 CC              BPL FORGLOP
435:    CC62 A0 07              LDY #7
440:    CC64 68                 PLA
445:    CC65 91 FB              STA (AKT),Y
455:    CC67 18                 CLC
460:    CC68 AD BA 02           LDA TEMP
465:    CC6B 69 08              ADC #8
470:    CC6D 8D BA 02           STA TEMP
475:    CC70 AD BB 02           LDA TEMP+1
480:    CC73 69 00              ADC #0
485:    CC75 8D BB 02           STA TEMP+1
495:    CC78 CE B9 02           DEC OSZAM
500:    CC7B 10 9F              BPL UJOSZLP
510:    CC7D AD BC 02           LDA TEMP+2
515:    CC80 85 01              STA #1
520:    CC82 58                 CLI
525:    CC83 60                 RTS

535:    02B5                    *=  $02B5
540:    02B5 00 20      KEP     .BYTE$00, $20
545:    02B7 18         SOR     .BYTE24
550:    02B8 27         OSZLOP  .BYTE39
555:    02B9 27         OSZAM   .BYTE39
560:    02BA 00 00 00   TEMP    .BYTE0,0,0
```

vetkezik be a maga 313 bájtjával.

A grafikus képernyő megjelenítésekor a C64 a színinformációkat az 1000 bájtos színtárból veszi, amely jelenleg a régi, a normál, a karakterek megjelenítésére szolgáló képernyő helyén van a $4Ø0—$7E7 (1Ø24—2Ø23)-ig terjedő területen. Ebben az üzemmódban a számítógép egy karakterhelynyi területen csak egyféle színt tud megjeleníteni a nagyfelbontású képernyőn. A színtároló egyes bájtjainak felső 4 bitje határozza meg a kigyújtott pont színét, míg az alsó 4 bitje pedig a ki nem gyújtott pontok, azaz a háttér színét adják meg. Ez egyúttal azt is jelenti, hogy ha a PRINT CHR$(147) utasítással letöröljük a normál képernyőt — azaz tulajdonképpen feltöltjük a SPACE, az üres hely kódjával, a 32-vel ($2Ø-val) —, akkor ilyenkor a fekete háttér előtt piros színű pontok jelennek meg a nagyfelbontású képernyőn.

A grafikus képernyő helye itt is 8192-nél ($2ØØØ-nál) kezdődik, 16191-ig ($3F3F-ig) tart, és 8000 bájtot foglal le a BASIC memóriaterületből. Emiatt változtatás nélkül csak 6 kbájtnál rövidebb program helyezhető el a BASIC munkaterületén anélkül, hogy maga a program bele ne lógjon a grafikus képernyőbe. Ilyenkor a grafikus képernyőn végzett műveletek módosíthatják a programot is! Ezt elkerülendő a BASIC munkaterület kezdetét célszerű feljebb tolnunk 16 kbájtra a következő utasítások segítségével:
POKE43,Ø:POKE44,64:

Az itt ismertetendő gépi kódú program a nagyfelbontású képernyőt forgatja felfelé bájtonként, de ez a bájtonkénti forgatás is csak egy pixelsornyi elfordulást idéz elő a grafikus képen. Ha a grafikus képet jobbra vagy balra forgatnánk vízszintes irányban, akkor létrejönne a grafikus képnek egy bájtnyi, azaz 8 bitnek (8 pixelnek) megfelelő elfordulása.

A gépi kódú programnak *(1. lista)* természetesen követnie kell a képernyő szerkezetét.

A grafikus képet a SYS(52224) utasítással forgathatjuk el felfelé egy pixelsornyival. Ilyenkor a kép legfelső, 40 bájtból álló grafikus képsora kerül a kép legaljára, és minden egyes grafikus képsor egy sorral feljebbre másolódik át a grafikus képernyőben.

Magyarázat a program soraihoz:

115—12Ø: A két nullás lapon lévő mutató segítségével végezzük a tényleges forgatást, az AKT az éppen aktuális bájtra, a KOV pedig az AKT alatt lévő (azaz a tőle számozásban 1 bájtnyira, vagy karaktersor váltásakor 313 bájtnyira lévő) bájtra mutat.

145—245: Átkapcsolás ROM-ról RAM-ra, a rendszermegszakítások letiltása és kezdőértékadás.

255—26Ø: A karakteroszlop legelső bájtját kimentjük: a veremtárba tesszük.

27Ø—31Ø: A két bájtmutató — dinamikus, állandóan változó — kezdőértéket kap. A kezdőérték nagysága a sorok és az oszlopok helyétől függ.

315—345: Az egy karakterhez tartozó bájtok forgatása itt jön létre. Ez a program egyik leglényegesebb része.

355—41Ø: A következő karakter legelső bájtját itt írjuk bele az előző karakter legutolsó bájtjába.

42Ø—425: Végére értünk-e az utolsó sornak is az adott karakteroszlopban?

435—445: A veremtárolóból visszaírjuk a karakteroszlop legelső bájtját az oszlop legutolsó bájtjába.

455—485: Az ideiglenes tároló értékét, amely a karakteroszlop legelső bájtjának a helyét tartalmazza, megnöveljük nyolccal.

495—5ØØ: Vége van-e az utolsó oszlopnak is a nagyfelbontású képernyőn?

51Ø—525: Az eredeti ROM/RAM konfigurációt visszaállítjuk, és az IRQ megszakításokat ismét engedélyezzük.

A gépi kódú program a tárban szabadon eltolható, áthelyezhető, mivel nem tartalmaz sem JMP utasítást, sem pedig a JSR utasítással végrehajtott szubrutinhívást — feltéve, ha a programban alkalmazott változók helyét (lásd az 1. lista 535—56Ø sorait) nem változtatjuk meg.

A *2. listán* a nagyfelbontású képernyőt forgató rutin BASIC betöltő programja látható. Ha ennek a BASIC programnak az 1Ø4Ø. sorában lévő W változóját megváltoztatjuk, akkor a gépi kódú programot a C64 memóriájának tetszőleges másik helyére is betölthetjük. Ebben az esetben a forgatást a SYS(W)-vel hozhatjuk létre.

2. lista

```
1000 FORI=693TO700
1010 READX:S=S+X:POKEI,X:NEXT
1020 IFS<>134THENPRINT"ADATHIBA":END
1030 DATA 0,32,24,39,39,0,0,0
1040 S=0:W=52224
1050 FORI=WTOW+131
1060 READX:S=S+X:POKEI,X:NEXT
1070 IFS<>17022THENPRINT"ADATHIBA":END
1080 DATA 165,1,141,188,2,169,52,120
1090 DATA 133,1,173,181,2,141,186,2
1100 DATA 173,182,2,141,187,2,173,184
1110 DATA 2,141,185,2,174,183,2,160
1120 DATA 0,173,186,2,133,253,173,187
1130 DATA 2,133,254,177,253,72,24,165
1140 DATA 253,133,251,105,1,133,253,165
1150 DATA 254,133,252,105,0,133,254,160
1160 DATA 0,177,253,145,251,200,192,7
1170 DATA 208,247,24,165,251,105,64,133
1180 DATA 253,165,252,105,1,133,254,160
1190 DATA 0,177,253,160,7,145,251,202
1200 DATA 16,204,160,7,104,145,251,24
1210 DATA 173,186,2,105,8,141,186,2
1220 DATA 173,187,2,105,0,141,187,2
1230 DATA 206,185,2,16,159,173,188,2
1240 DATA 133,1,88,96
```

3. lista

```
5 REM EZ A PROGRAM A $2000-NEL LEVO
10 REM NAGYFELBONTASU KEPERNYOT
15 REM FORGATJA FELFELE BAJTONKENT
17 REM AZAZ EGYUTTAL BITENKENT
20 POKE53265,PEEK(53265)OR32
25 POKE53272,PEEK(53272)ORØ
35 FORI=1024TO2023:POKEI,224:NEXT
45 FORI=8192TO16192:POKEI,.:NEXT
55 FORJ=320TO7360STEP320
60 FORI=ØTOØ200:POKEØ192+I+J,7:NEXT:NEXT
65 FORI=8195TO16192STEP8:POKEI,.:NEXT
70 FORJ=ØTOØ199:SYS(52224):NEXT
71 BASE=8192:KSØR=5:KOSZLOP=5:SØR=16:OSZLOP=16
73 KEP=BASE+KSØR*320+KOSZLOP*8
75 HK=INT(KEP/256):LK=(KEPANDØ255)
77 POKE693,LK:POKE694,HK
78 POKE695,SØR:POKE696,OSZLOP
79 FORJ=ØTO(SØR+1)*8-1:SYS(52224):NEXT
80 POKE53265,PEEK(53265)ANDØ223
85 POKE53272,PEEK(53272)ANDØ247
87 LK=Ø:HK=32:SØR=24:OSZLOP=39
88 POKE693,LK:POKE694,HK
89 POKE695,SØR:POKE696,OSZLOP
90 PRINT"":END
```

A *3. listán* látható BASIC program pedig a gépi kódú rutin működését mutatja be. (Figyelem! Ennek a programnak a működése előtt a C64-es memóriájába kell tölteni a gépi kódú forgató rutint!) A 3. lista programja először egy grafikus

képet hoz létre a nagyfelbontású képernyőn, azt egyszer felfelé teljesen körbeforgatja, majd pedig a kezdőértékek megfelelő megválasztásával lecsökkenti a forgatandó memóriaterületet, és a kép kezdetét is máshol jelöli ki. Ezután ezt a kisebb területű képet forgatja felfelé a nagyfelbontású képernyőn, miközben annak a forgatás által nem érintett része változatlan marad.

A POKE 693, LK : POKE 694, HK segítségével változtathatjuk meg a forgatandó képterület kezdőcímét, ahol LK a kép kezdőcímének alsó bájtja, HK pedig a felső bájtja. Így a SIMON'S BASIC által létrehozott képek — amik $E000 fölött helyezkednek el — akkor forgathatók, ha LK=Ø (=$ØØ) és HK=224 (=$EØ).

Felhívom a figyelmet arra, hogy SIMON'S BASIC-ben a képhez tartozó 1000 bájtos színtároló helye $CØØØ-tól $C3FF-ig tart (49152—5Ø175), emiatt ezen a tárolóterületen nem szabad gépi kódú programokat elhelyeznünk, mert különben a SIMON'S BASIC grafikai utasításai tönkreteszik a gépi kódú programokat. A POKE 695, SOR : POKE 696, OSZLOP utasítások segítségével a forgatásban részt vevő (SOR+1) × (OSZLOP+1) számú karakternek megfelelő területre korlátozhatjuk a forgatás hatását. A SOR-ra és az OSZLOP-ra a következő egyenlőtlenségnek kell teljesülnie:

O ≤ SOR ≤ 24
O ≤ OSZLOP ≤ 39

Végezetül pedig próbáljuk ki a következő módosítást: POKE 52267,169: POKE 52268,Ø

Ez után az utasítás után, a forgató gépi kódú rutin már nem forgatja körbe a grafikus képet, hanem a kép mintegy „kiúszik" a képernyőről, mert a grafikus képernyő memóriatartalma törlődik. Ez az utasítás az 1. lista 255. sorában található: LDA (KOV),Y-t változtatja át LDA#$ØØ-ra.

A következő alkalommal a nagyfelbontású képernyő vízszintes irányú, bájtonkénti forgatásáról lesz szó, amivel majd le is zárjuk a forgatásról szóló sorozatunkat.

SZABÓ PÉTER PÁL

MINDENT A VEVŐÉRT?

A nagy nyugati szoftverházak közül a Microsoft volt az első, amely valódi, élő kapcsolatokat épített ki magyar cégekkel, és csaknem teljes termékskálájával jelen van a magyar piacon. Ezzel együtt a cég magyarországi eladásai az elmúlt évben még csak mindössze 0,75 millió nyugatnémet márka értékűek voltak — úgy tűnik, jócskán vannak még kiaknázandó lehetőségeik.

Amikor a szoftverházakat „felosztották" egymás között a magyar cégek, a Microsoft hazai partnere a Számalk, az Ashton-Tate-é a Novotrade Rt., a Borlandé a Softinvest lett. Mindmáig csak az első páros együttműködése tűnik teljesen megalapozottnak, a másik két esetben alig vannak nyomai az üzleti kapcsolatoknak.

Kelet-Európában a Microsoft cég ügyeit a Microsoft GmbH Deutschland intézi, a szocialista országokban Magyarországon (a Számalkon) kívül van már partnerük Bulgáriában, Jugoszláviában és a Szovjetunióban (Dialog, Elorg). Lengyelországban az ELWRO, az NDK-ban pedig a Robotron a lehetséges jövőbeli üzlettársuk.

Bár eredetileg nem akartak teljes forgalmazási jogot adni egyetlen magyar cégnek sem, végül is az alkalmazási programok területén ez a megoldás bizonyult életképesnek. Az MTA SZTAKI Cosy, amely tavaly még forgalmazott háromféle Microsoft-szoftvert, idén már nem vállalkozott rá. Ennek oka valószínűleg a kevés devizában keresendő, csakúgy, mint a Műszertechnika Kisszövetkezetnél, amely tíz darab MS—Windows 1.03 angol nyelvű változatának beszerzését tudta megoldani saját keretéből. A kereskedelmi osztály információja szerint eddig legfeljebb ötöt értékesítettek belőle — darabját 40 000 forintért. A Magiszter bizományosként értékesít egy-egy régebbi változatot (MS—COBOL 2.0 — 100 000 forint, MS WORD 3.0 — 87 500 forint).

Az Econorg a Microsoft GmbH Deutschlanddal kötött együttműködés keretében jogosult néhány Microsoft-termék forgalmazására. Így nyílott lehetőségük tíz darab Windows 2.0 és C 5.0 fordítóprogram megvásárlására, a kettőt együtt 160 000 forintért adják. Az Econorg kifejlesztette a Windows nyomtatási segédletét az IBM Quietwriter 3-hoz, ezt a Microsoft bevizsgálta, és az Econorg nyugatnémet vegyesvállalatnál már árusítják is. A legtöbben (Econorg, Könyvértéka, Számalk) a Multiplan-üzletben — így a magyar nyelvű változat forgalmazásában — vesznek részt, s bár mutatkozik némi érdeklődés, átütő piaci sikerről nem beszélhetünk.

Reiner Michl, a kelet-európai üzletág vezetője és Christian Wedell vezérigazgató egy nyári mini-sajtótájékoztatón elmondta, hogy szoftvereik magyarországi elterjedésének talán legfőbb korlátja a számítástechnikai kultúra jelenlegi állapota. Ezért is keresik a lehetőséget egyes termékek (Multiplan — Green Data AG) magyar nyelvű segédleteinek elkészítésére.

Teljes jogú partnere a Microsoftnak Magyarországon csak egy van: a Számalk menedzser üzletága. A tőlük megrendelt Microsoft-terméket egy hónapon belül szállítják. A szoftverforgalmazás nagyságára jellemző, hogy a magyar termékek 250-es csúcsával szemben egy-egy nyugati termékből 50-100 programcsomag eladása is nagy sikerként könyvelhető el. A Számalk saját forrásból fedezi a Microsoft-termékek behozatalát, az árképzésnél pedig a mindenkori lehetőségekhez igazodik. Tervezik a szoftverforgalmazáshoz kapcsolódóan az oktatási és menedzserképzési tevékenység kibővítését is. Gond lehet az együttműködésnél a forgalmazandó mennyiséggel: a Számalk szándéka ésszerű kompromisszumot kötni — a magyar piac lehetőségeihez igazodva elégedjék meg a gyártó kisebb, például 100 darabos eladásokkal is. A magyar cég saját vidéki hálózatát kívánja kiépíteni mintegy 30—-70 helyen, de ezenkívül ún. dealer-hálózatot is működtet majd a Microsoft-termékek forgalmazására. A magyarországi árak kialakításába a Microsoft GmbH nem szól bele.

Leginkább az operációs rendszerek magyarországi forgalmazásában érvényesül a Microsoft eredeti piaci elképzelése: szinte valamennyi hardvergyártó számára (Percomp Társulás, Videoton, SZKI) megteremtik annak a lehetőségét, hogy hozzájussanak az operációs rendszerekhez. Valamint azt is, hogy saját maguk állítsák elő és forgalmazzák az amerikai változatot.

J. V.

Fényújság FORTH-ban

A fényújságot először BASIC-ben próbáltam megírni C16-os gépen, de amikor a sebességgel és programstruktúrával meggyűlt a bajom, áttértem a FORTH-ra és a C64-re. Az alkalmazott FORTH rendszer a Microproducts cég terméke. Ezt is C64 FORTH-nak hívják, de sokban különbözik az Abersoft terméktől.

A fényújság mindössze hat screent foglal le. A rendszer megfelelő módosításával mintegy huszonöt képernyőt tud váltogatni. Mindehhez felhasználhattam a FORTH saját full-screen szövegszerkesztőjét. Így végül két nap alatt elkészültem. Egyedül az hátráltatott a munkában, hogy a szövegszerkesztő nem engedi meg a grafikus karakterek használatát. Tapasztalt FORTH programozók természetesen több felesleges cicomát találhatnak a programban, de az esetleges módosítás már mindenkinek a magánügye. Lássuk a programot, ami a 25. screenen kezdődik.

A K-MAX konstans az utolsó lap száma, a K-MIN az első lapé. Az 1. screen a lapok időadatait tartalmazza.

A megjelenített lapok számát célszerűen a K-MAX konstans változtatásával állíthatjuk be. Az ARRAY szó egy egészeket tartalmazó tömböt hoz létre. Két ilyen tömbre van szükségünk. A KEP nevűben a lapokhoz rendelt időtartamot (hány másodpercig legyen látható, Ø esetén kikapcsolva), az ADDRESS nevűben pedig az inicializálás során betöltött screenek címeit tárol-

```
SCR # 25
( * FENYUJSAG ************* SCREEN# 1 * )
FORTH DEFINITIONS DECIMAL

22 CONSTANT K-MAX      2 CONSTANT K-MIN
K-MAX K-MIN - CONSTANT K-SUB
0 VARIABLE IDO         0 VARIABLE PIC-P

: ARRAY (BUILDS 2 * ALLOT
        DOES> SWAP 1- 2 * + ;

K-SUB ARRAY KEP
K-SUB ARRAY ADDRESS
: ARR-INIT K-SUB 2 * ERASE ;
1 KEP ARR-INIT         1 ADDRESS ARR-INIT
: ALLITAS -1 SWAP ! ; : PAGE 147 EMIT ;
: SEC SWAP ! ;

: INIT
  PAGE CR CR ." FENYUJSAG " 1 LOAD
  K-MAX K-MIN
  DO
    I BLOCK 40 + ! K-MIN - 1+ ADDRESS !
  LOOP PAGE
; -->
```

```
 OK

SCR # 26
( * FENYUJSAG ************* SCREEN# 2 * )

( ------------------------------------ )
( CASE STATEMENT *** CH. E. EAKER      )

: CASE ?COMP CSP @ !CSP  4 ; IMMEDIATE

: OF 4 ?PAIRS COMPILE OVER COMPILE =
  COMPILE 0BRANCH HERE 0 , COMPILE DROP
  5 ; IMMEDIATE

: ENDOF 5 ?PAIRS COMPILE BRANCH HERE 0 ,
  SWAP 2 [COMPILE] ENDIF 4 ; IMMEDIATE

: ENDCASE 4 ?PAIRS COMPILE DROP
    BEGIN SP@ CSP @ = 0=
    WHILE 2 [COMPILE] ENDIF REPEAT CSP !
; IMMEDIATE
( ------------------------------------ )
-->

SCR # 27
( * FENYUJSAG ************* SCREEN# 3 * )
: WRITE-MENU ( FUNCTION --- FUNCTION )
  19 EMIT CR CR CR CR CR CR
  5 1 DO ."                "
      DUP I = IF 10 EMIT ENDIF
    I CASE 1 OF ."    KILEPES     " ENDOF
           2 OF ."   SZERKESZTES   " ENDOF
           3 OF ."  IDOK ALLITASA  " ENDOF
           4 OF ."      FUTAS      " ENDOF
      ENDCASE
      DUP I = IF 146 EMIT ENDIF CR
      LOOP CR CR CR
." ELORE F1, HATRA F3, VALASZT RETURN "
;
: MENU
  0 1
  BEGIN WRITE-MENU 19 EMIT KEYIN CASE
    133 OF 1+ ENDOF 134 OF 1- ENDOF
    13  OF SWAP DROP -1 SWAP ENDOF
    ENDCASE DUP 1 < IF DROP 4 ENDIF
    DUP 4 > IF DROP 1 ENDIF
  OVER UNTIL
  SWAP DROP ; -->



SCR # 28
( * FENYUJSAG ************* SCREEN# 4 * )
: 0->BL
  BEGIN DUP @ WHILE 1+ REPEAT BL SWAP !
;
: SZERKESZT
  BEGIN
    PAGE CR CR CR CR
."         HANYAS SZAMU KEPERNYOT 1..20 "
    PAD 3 EXPECT PAD 0->BL
    PAD 1- NUMBER DROP DUP DUP
    K-SUB 1+ < SWAP 0 > AND
  UNTIL
  1+ EDIT ;
: IDOK 0 IDO ! 1 EDIT IDO @ 0=
    IF 1 LOAD ENDIF ;
: 1SEC 7000 ! DO LOOP ;
: VAR 0 SWAP 0  0  ( FLAG SECS 0 )
  BEGIN 2DUP >
  WHILE ?TERMINAL
    IF DROP SWAP DROP -1 SWAP DUP
    ELSE 1+ 1SEC ENDIF REPEAT 2DROP ;
-->

SCR # 29
( * FENYUJSAG ************* SCREEN# 5 * )
: HEADER PAGE 2 SPACES
." --- FENYUJSAG ----------------------"
CR ;
: PIC-1 1 PIC-P ! ;
: FUT PIC-1
  BEGIN BEGIN PIC-P @ KEP @ 0= ?TERMINAL
        NOT AND
        WHILE 1 PIC-P +! PIC-P @ K-SUB >
          IF PIC-1  ENDIF  REPEAT
  HEADER
  PIC-P @ KEP @ 0= IF -1 ELSE
    PIC-P @ ADDRESS @
    19 EMIT 17 EMIT 920 TYPE PIC-P @ KEP
    @ VAR ENDIF
  1 PIC-P +! PIC-P @ K-SUB > IF
      PIC-1  ENDIF
UNTIL ;
-->

SCR # 30
( * FENYUJSAG ************* SCREEN# 6 * )
: TXT INIT
    BEGIN PAGE MENU PAGE CASE
      1 OF FLUSH QUIT   ENDOF
      2 OF SZERKESZT    ENDOF
      3 OF IDOK         ENDOF
      4 OF FUT          ENDOF
      ENDCASE
    AGAIN
```

juk. Az IDO változóra és az ALLITAS szóra a 1. screen feldolgozásához van szükség, szerepét később látjuk.

Az INIT szó betölti az első screent (ezzel beállítva az időket!), majd beolvassa a képernyőket és az ADDRESS tömbben tárolja a címeket.

A 26. screenen a CASE szerkezet megvalósítását láthatjuk. Ha valamelyik FORTH rendszerben ismert ez a szerkezet, ott elhagyható.

A 27. screen tartalmazza a menü kiírását végző WRITE-MENU szót és az ezt vezérlő MENU szót. A veremben mélyebben levő szám jelzi, hogy a választás megtörtént, a felső szám pedig a kiválasztott funkciót. A WRITE-MENU ez alapján a megfelelő funkciót inverz módban írja ki. A funkciót a (RETURN)-nel aktivizálhatjuk. A MENU végül a választott funkció kódját hagyja a veremben. A lehetséges funkciók neveit a WRITE-MENU szóban találjuk.

A 28. screen a Ø→BL szóval kezdődik. Ez egy címet vár a vermen, megkeresi a fölötte levő első Ø bájtot, és kicseréli a szóköz kódjára. Ez a szó csak azért került be a programba, mert nem boldogultam a (NUMBER) szóval — úgy tűnik, hibásan került a rendszerbe, vagy legalábbis nem a standard szerint működik —, a NUMBER viszont a konvertálandó sztring végén egy szóközt vár.

Ezután a SZERKESZT szó következik, amely egy lap szerkesztését vezérli. Bekéri egy lap (screen) számát, és átadja a vezérlést a FORTH editorának.

A következő az IDOK szó. Ez az 1. screent szerkeszti, majd megvizsgálja, hogy az IDO változó Ø-e. Ha nem, a szerkesztőből kilépés során betöltést is kértünk (CTRL+L), így feleslegesen nem tölti be újra a screent. Ez a megoldás erősen vitatható és nem is bolondbiztos, de valószínűleg ez a legkevésbé nehézkes.

Következik az 1SEC szó. Ez körülbelül egy másodpercig vár.

Végül a VAR szó található ezen a screenen. A vermen adjuk meg, hogy hány másodpercig várjon. A RUN/STOP billentyűvel megszakíthatjuk ennek a szónak a futását, ekkor a vermen −1-et ad vissza, egyébként Ø-t.

A 29. screen első szava a HEADER, funkciója egyértelmű: ez írja ki a képernyő első sorába a fejlécet. Ezután a PIC-1 szó következik, amely a PIC-P változót 1-re állítja.

A FUT szó a képernyőnek futáskori vezérlését látja el. A belső ciklus kiválasztja a következő bekapcsolt képernyőt, az utána levő rész pedig kiírja. Az utolsó sor szintén üresen marad, különben a C64 egy sorral följebb tolja a képernyőt!

Ha az összes képernyő ki van kapcsolva, a belső ciklus végtelen lesz, amit a RUN/STOP-pal tudunk megszakítani. A külső ciklus kilépési feltétele a vermen levő szám, amit a VAR szó tesz le, tehát a külső ciklus addig fut, amíg a VAR szó közben nem nyomjuk meg a RUN/STOP-ot.

Végül a 30. screen. Ez a teljes rendszer vezérlését tartalmazza, a TXT szót. Működése egyértelmű: a MENU szó leteszi a veremre a kiválasztott funkció kódját, a CASE pedig meghívja a szükséges szót. Kilépés előtt elmenti a módosított lapokat.

Szólnom kell még a pufferekről. Ez a FORTH alaphelyzetben 5 blokkpuffert tartalmaz. Ez kevés, hiszen futás közben állandóan töltögetne, ami külön nyilvántartást igényelne. Ezért át kell írnunk a FIRST rendszerkonstanst és a PREV változót. A 31. screen ezt a módosítást mutatja, amit közvetlenül a FORTH indítása után töltsünk be, és utána adjuk ki a COLD utasítást! A RECONFIG úgy számítja ki a 'FIRST és a PREV új értékét, hogy még 20 puffernek legyen hely. A RECONFIG hívása után álló 'FIRST! megoldás egy kissé illegális, de tökéletesen működik.

Miután a teljes rendszert betöltöttük, a SAVE-SYSTEM TXT utasítással lemezre menthetjük a fényújságot, amely így új, önálló FORTH szótár lesz.

A program nyilvánvalóan nem tökéletes. Több helyen hiányos az ellenőrzés. Ha a szövegszerkesztőből CTRL+F-fel lépünk ki (FLUSH), csúnya bonyodalmak léphetnek fel stb., de a cél nem atombiztos, hanem rövid és egyszerű program volt.

TÖRÖK TIBOR

Víz-kvíz

Ki gondolná, hogy egy helyi vízműben a döntésekhez, végső soron a zökkenőmentes vízellátáshoz milyen sok adat és információ szükséges?! S ha már egyszer nélkülözhetetlenek, márpedig azok, akkor az sem mindegy, hogyan bánnak azokkal, mivelhogy a vízmű működésében szinte minden pillanat és mozzanat sorsdöntő lehet. Simon Gábor és Vágó Bálint a Dunántúli Regionális Vízmű Nyugat-balatoni Üzemigazgatóságának Központjában éppen az iménttektől ösztönözve egy olyan megfigyelési rendszert dolgozott ki Commodore gépekre, amely a vízmű munkájában új korszakot nyitott. Azért, hogy a számítógépes rendszer könnyebben érthető legyen, mindenekelőtt tekintsük át röviden a vízmű felépítését.

A bauxitbánya egy központi medencébe szivattyúzza az ivóvizet, ahonnan az a regionális vezetékbe áramlik. A központi vezeték hatvan kilométeres körzetben, kilenc csomópontban ágazik el. Az elágazásoknál kisebb tározómedencék és szivattyútelepek vannak. A csomópontokban elektronikus műszerek mérik a vízszintet, a víz mennyiségét, a tolózárak nyitottságát, a szivattyúk üzemállapotát, valamint nagyon sok más adatot és állapotot. Minderre a vízmű megfelelő működtetéséhez szükség van, de mindezt egyszerre áttekinteni lehetetlen. A helyzetet az is nehezíti, hogy a csomópontok és a központ között nincs kábelösszeköttetés, aminek a kiépítése különben nagyon költséges lenne. Maradt tehát a rádió-összeköttetés.

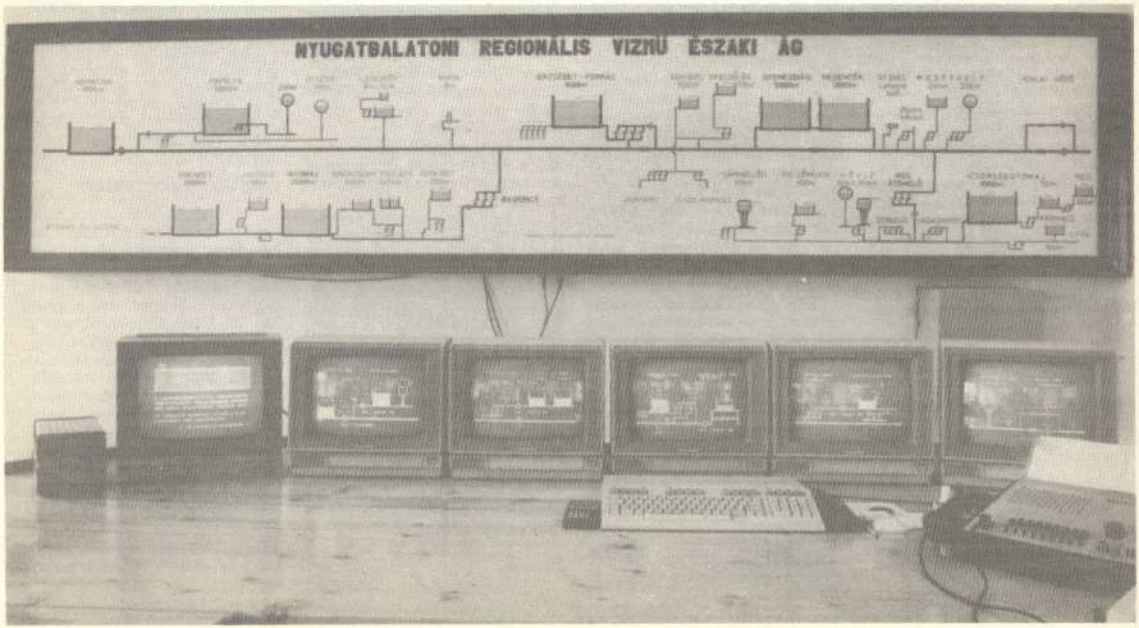
A vízmű egyébként folyamatosan fejlesztette a telemechanikai rendszerét. Ennek eredményeképpen olyan távközlést hoztak létre, hogy az adók állandóan sugározzák a mérési eredményeket. Ezeket azonban a döntések előtt, a munka precíz irányításához értékelni kell. Erre született az ötlet, majd a megoldás is, hogy ezt a feladatot a számítógépekre bízzák.

A rendszer lelke a jelmechanikai dekóder, ami a vevőkészülék által észlelt jeleket digitalizálja és alakítja át számítógépek számára érthető kódokká. A rendszer az adókat folyamatosan és sorban lekérdezi. Egy ilyen kör három percig tart. A kódokat csomópontonként egy-egy C64-es gép értelmezi. Ezek állandóan gépi kódú programmal működnek, amely feldolgozza a kapott adatokat, és a telep aktuális állapotát kijelzi a monitorra. A kódokat úgy szerkesztették meg, hogy ha a jel valamilyen okból torzul (például légköri zavar miatt), azt a számítógép felismeri, és nem fogadja el a helytelen információt. A C64-eket egy központi C128-cal kötötték össze. A diszpécser ennek segítségével „lapozhat" az információk között, és ki is nyomtathatja adatok formájában a telepek állapotát. Így utólag is megnézhető, hogy adott időpontban hogyan működött a vízmű.

A Commodore-ok kapacitása közel sincs kihasználva. A fejlesztők azért döntöttek az elosztott intelligenciájú rendszer mellett, mert ha az egyik gép meghibásodik, a többi csomópontról érkező információ még értékelhető. A telemechanikai rendszer igen megbízható, igaz, azon az áron, hogy viszonylag lassú.

Tekintettel arra, hogy a gépek szabadon programozhatók, az ötlet majdhogynem bárhol alkalmazható. Kapásból néhány lehetőség a távellenőrzésre: távfűtés szabályozása, felvonók figyelése, vagyonvédelem, árvízvédelem, üvegházak hőmérsékletének ellenőrzése. Az alkalmazások köre szinte kifogyhatatlan!

PINKE GYÖRGY



A HIÁNY HIÁNYT SZÜL EGY GÉP

1988. június. Az ország különböző pontjain 12 ezer TVC van, ennek nagy része iskolákban. A gép olcsó és közkedvelt, a programok száma viszont kevés, és akad más kisebb-nagyobb hiba, kifogásolnivaló is. Mi hát mindezek oka, mi a valóság, s mi igaz a szóbeszédből? Elsősorban a tulajdonosok levelei nyomán indultunk útnak az illetékes cégekhez. Aztán az eredetileg egyszerűnek tűnő „nyomozás" érdekes fordulatokat vett.

Először Budapesten, a Videoton Számítástechnikai Kereskedelmi és Szolgáltató Leányvállalatánál jártunk, ahol Gergely János és Kiss Pál fogadott minket szívélyesen.

— Elterjedtek olyan pletykák, melyek szerint a TVC egy régebbi Enterprise gép licence. Tényleg így van?

— Ebben csupán annyi az igazság, hogy mindkettőt ugyanaz az angol cég tervezte.

— Pontosan mióta gyártják a TVC-t, s milyen a kereslet iránta?

— A sorozatgyártást 1985-ben kezdtük el, azóta majdnem 3000 darab 32 k és 9000 darab 64 k gép talált gazdára, illetve van kereskedelmi forgalomban. A vásárlók elsősorban az iskolák, hiszen iskolaszámítógépnek is szántuk, de bőven akadnak magánszemélyek is, akik főként a gép tudásához viszonyított olcsó ára miatt vesznek ilyen típusú számítógépet.

— Hogyan lehetett azt elérni, hogy a gép ára ne emelkedjen túlzottan?

— Igaz, hogy az árra rá kellett tenni az általános forgalmi adót, viszont máshol a gyártás folyamán csökkenteni tudtuk a kiadásokat, s ez eredményezte a csekély mértékű áremelést.

— Ez mindenesetre örvendetes, ám a géppel szemben problémák is felmerültek. A beépített botkormány labilis, hamar tönkremegy, ezenkívül a reset gomb sem tökéletes.

— Ezekről mi is tudunk, de sajnos, megváltoztatásukkal kapcsolatban még nem született döntés. Erről talán a székesfehérvári fejlesztők többet tudnak mondani.

— Ugyanez a helyzet a kitámasztó lábakkal is?

— Nem, az újabb gépek már stabil lábakkal készülnek.

— És az a billyentyű, amelyik backlash helyett aposztrófot ír?

(Erre mindketten nyomogatni kezdik az előttük levő gép billentyűit, próbálják kinyomtatni újra és újra, de a backlash csak nem akar a helyére kerülni.)

— Erre azt hiszem, nem tudok válaszolni, ez még számomra is újdonság.

— Hallottunk panaszokat a lemezegységgel kapcsolatban is. Várható-e új lemezegység megjelenése?

— Igen, már el is készült néhány egy- és kétmeghajtós egység. A régi UPM kompatibilis volt. Az új VT-DOS kompatibilis, de képes lesz elolvasni az elődje által készített lemezeket is. Ennek nagy előnye, hogy egy cartridge segítségével használhatóvá válik a CP/M operációs rendszer a 64K + és a 32 k-s gép kibővített változatán. (Erről az újdonságról legutóbbi lapszámunkban közöltünk részletesebb információt.)

— És mit tudnak tenni azok a felhasználók, akiknek 64 k-s régi gépük van?

— Természetesen át lehet majd alakítani ezeket a gépeket is. Valószínűleg a szervizek is elfogadják a javaslatunkat, s vállalják majd az átalakítást, hiszen csupán egy IC beültetéséről van szó.

— Mely szervizek foglalkoznak a javítással?

— A TVC-vel több szervizlánc is foglalkozik. Az iskolai gépeket a Professional Országos Számítógép Szerviz, a magántulajdonúakat az Agroindustria Innovációs Vállalat javítja.

(Kaptunk egy listát, amelyen további nyolc cég neve és címe olvasható. Ezzel, úgy érezzük, a szervizelés gondja tényleg megoldottnak tekinthető. Minket azonban nemcsak ez érdekelt.)

— Egy gép megítélésében egyik legfontosabb szempont a szoftver. Milyen a programellátottság?

— Ez igen összetett kérdés. A tulajdonosok mindig több és több programot szeretnének, ami miatt a kínálat lemaradozik. Meg kell azonban mondani, hogy sajnos ennek más oka is van. Természetesen a Videoton is készített jó néhány programot, de született egy szerződés a Novotrade-dal, amely szerint ők a 2,6 millió forint értékű gépek, valamint a 400 ezer forint reklámköltség fejében vállalják hardverkiegészítők, évi száz program és négy könyv elkészítését. Most 1988 júniusa van, harminc programmal és két könyvvel.

— Mi volt erre az önök reakciója?

— Sajnos, nem tehettünk semmit, mivel a szerződés szankciókat nem tartalmaz.

— És ez miért van így?

— A Novotrade a magyar piacon monopolhelyzetben van. De találtak ürügyet is. Azzal védekeznek, hogy a gépeket és dokumentációkat két hónapos késéssel, 1987 februárjában szállítottuk le részükre. Azóta eltelt másfél év, s programokat, könyveket még ma is alig lehet látni. Hivatkozásuk alapja most a kazettahiány.

— De hát sorra látni a Novotrade-os kazettákat és a hozzájuk kapcsolódó hirdetéseket, úgyhogy kazettahiányról ilyen értelemben aligha beszélhetünk.

— Sajnos, nemcsak a kazetták számával, hanem a minőségükkel is baj van. A szalagok könnyen nyúlnak, hamar használhatatlanná válnak.

— Hol lehet ezeket egyáltalán beszerezni?

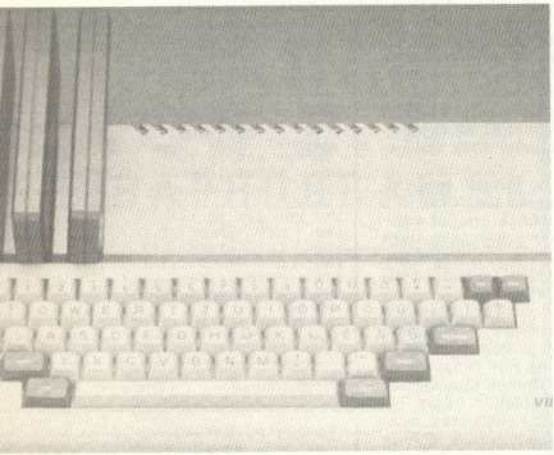
— Elvileg a Centrum áruházak foglalkoznak árusításukkal, ám ők előtérbe helyezik az Enterprise gépek és programok forgalmazását. A TVC csak ezután jön. Ez mindenképpen késlelteti a TVC 64 K + megjelenését is, ami üzletpolitikai okok miatt majd csak akkor kerülhet a pultokra, ha az elődje elfogyott.

Haszontalan reklám

Utunk ezek után Székesfehérvárra vezetett, ahol kissé körülményesen, a bürokrácia által kiszabott utakat megjárva ugyan, de bejutottunk a gyárba. Itt Cseh Tibornak, aki a Számítástechnikai Gyár fejlesztési osztályán dolgozik, az előbbiekhez hasonló kérdéseket tettünk fel. Ő ugyan bizonyos dolgokról csak magánemberként mondott véleményt, de volt némi tanulság a válaszaiban.

— Olvasóink panaszolták a reset gomb és a beépített botkormány gyenge minőségét, s a billentyűzet egy apró hibáját. Lehetne ezeken valamit javítani?

SZÁZ BAJT CSINÁL?



— Természetesen ezeket házilag bárki kicserélheti, de mi a gyártáskor arra törekszünk, hogy a gép minél olcsóbb legyen, hiszen magasabb áron valószínűleg kisebb lenne a kereslet. A botkormány nyomókeresztje egyébként most már erősebb anyagból készül, de ha mégis eltörne, a garancia erre is vonatkozik. A reset gomb kicserélése többek között azért sincs betervezve, mert a gép dobozát előállító szerszámot kellene módosítani, ami pedig ugyancsak drága. A billentyűzet szépséghibája a festéshez szükséges maszk pontatlanságából ered, de a gépkönyv végén található egy helyesen kinyomtatott kiosztás, ami garantáltan pontos.

— Szerintünk a gépnek nagyobb reklámra volna szüksége. Mi erről a véleménye?

— A Videotonnak nem érdeke a gép reklámozása, ezen nekünk semmi hasznunk nincs; a TVC-t ugyanis körülbelül ugyanolyan áron állítjuk elő, mint amennyiért adjuk. Mi állami nagyvállalat vagyunk, fentről kapjuk az utasítást és néha a támogatást. Ezek után nem hinném, hogy a vállalat érdekei nagyobb propagandát kívánnának meg.

— És mi a helyzet a TVC-klubokkal?

— A Videoton a klubjainak csak gépeket ad. Én személy szerint csupán két TVC-s klubról tudok, az egyik itt Székesfehérváron, a Videoton Oktatási Központban, a másik Budapesten, a Marcibányi téri művelődési házban található. (A budapesti klubot időközben kitették innen, jelenleg a Bem rakparton találtak otthont. — A szerzők.)

Az ördög sohasem alszik?!

Természetesen kíváncsiak voltunk a Novotrade hozzáállására is. Csak kisebb kutatással tudtuk meg a TVC-ügyben jártas emberek nevét, de sajnos, a szerződés ügyében illetékes nem került elő. Szerencsére azonban más jellegű információkat bőven kaptunk az Octasoft-ban. Első kérdésünk a kazettahiánnyal volt kapcsolatos.

— Sajnos, az utóbbi időben tényleg nagyon nagy kazettagondokkal küzdöttünk, de remélem, hogy ez már a múlté. A szocialista importban semmiféle szankcióra nincs lehetőségünk, aminek következménye, hogy a megrendelt kazettákat csak néha-néha érkeznek meg. Ráadásul egy másik Enterprise-os megrendelésnél már kötbért is fizettünk, így annak a sorozatnak az előállítása fontosabb volt, ennélfogva az élvezett előnyt. Az áruházakban pedig azért szembetűnőbb az Enterprise programok magasabb száma, mert azokat a Centrum már korábban felvásárolta. A TVC programokra viszont jóval kisebb a kereslet.

— De a minőséggel is vannak problémák.

— A kazetta ma már nem a legmodernebb adathordozó, ennek ellenére még szükség van rá, s ha használjuk, legalább a magnetofon lejátszófejének pontos beállítására ügyeljünk. Ugyanazt a programot az egyik magnón be lehet tölteni, a másikon nem. Egyébként már mi is szerettünk volna venni egy modernebb másológépet, de ez a lehetőség anyagi okok miatt kútba esett, mert az oktatóprogram nem a legkifizetődőbb.

— Egyáltalán hány program kapható a Novotrade forgalmazásában?

— Jelenleg körülbelül negyven. De mutatok egy megrendelőt, ez ha jól tudom, teljes.

— Ez elképesztő, ebből a 39 féle programból összesen csak 77 darabra van megrendelés?

— Ettől a cégtől igen, de a többi sem különb. Szeretnénk nagyobb tételben is eladni iskoláknak, hogy az árakat lejebb szoríthassuk. Állami támogatás inkább csak a gépekre van, s a programok vásárlására kevés pénz jut.

— Pedig ha nincs program, a gépet sem veszik annyian.

— Ez már csak egy ilyen ördögi kör. Mi olcsón behoztuk volna továbbra is a C Plus/4-et, saját valutakeretből, de a belföldi ipar tért hódított. Mire kezdtünk belejönni a Plus/4-be, stílust kellett váltanunk. De ha csak ezen múlott volna! Jártam én már olyan számítástechnika-oktatási tanácskozásokon is, ahol . . . De ezt hagyjuk.

Gép — kazetta — könyv. Három egység, amely nélkülözhetetlen. Vajon mi lehet a legsanyarúbb sorsúval, a harmadikkal, a könyvvel? Erről a Novotrade-nél telefonon érdeklődtünk. Kérdéseinkre Szemere Gabriella adott választ.

— Szeretnénk tudni, hogy az eddig megjelent kiadványokon kívül (TVC BASIC, Operációs rendszer) számíthatunk-e újabbakra?

— Igen, most jelenik meg egy hardverről szóló kiadvány, és további két könyv is szerepel terveinkben, a gépi kódú programozásról és a TVC ROM-ról szóloak, melyeket az év végéig szeretnénk megjelentetni.

Mindenkinek kára . . .

Az eddigiek mindenképpen azt sugallják, hogy a Novotrade és a Videoton kapcsolatai nem nevezhetők rózsásnak. Erre konkrét példa, hogy a gépre eddig a hivatalosan forgalomba került programokról sehol sem tudtak összefoglaló kiadványt adni. A két cég szerződésében szó van egyébként hardverkiegészítőkről is, holott a Novotrade-esek szerint cégük nem is foglalkozik ilyesmivel, továbbá a szerződés nem köti ki ezek számát sem, tehát akár semmibe is vehető.

Szomorú, de végül a boltokban tapasztaltak szerint sem a legjobb a programkínálat. Sehol sem haladta meg a hatnyolc darabot.

Ez hát a keserű igazság, s mindennek kárát a felhasználó vagy az a diák látja, akinek az iskolában ilyen gépparkon kell tanulnia. De mi van akkor, ha nem csak kell, hanem meg is szeretné tanulni szegény diák a számítástechnika fortélyait, vagy a számítógép segítségével szeretné elsajátítani más tantárgyak ismeretanyagait? Hiszen azért van a gép, hogy vele tanuljunk, vagy legalábbis azért lenne. Vagy netán valami porszem került az iskolaszámítógép-program gondosan megolajozott fogaskerekei közé?

BÁRTFAI BARNABÁS—
VÁMOS SÁNDOR

SZÁMÁBRÁZOLÁS ÉS -TÁROLÁS

Az általános iskola alsó tagozatától kezdve a különböző szintű számítástechnikai tanfolyamokig lépten-nyomon találkozhatunk a különböző számrendszerekkel.

A számítástechnikai folyóiratokban, a BASIC példatárakban se szeri, se száma a számrendszerek közötti átváltást lehetővé tevő programoknak. Persze amióta a Commodore 3.5 verziójú BASIC-jével az ilyen átváltás a DEC és a HEX$ függvényekkel megvalósítható, azóta vesztett a varázsából a számkonverziós programok írása.

Az Enterprise-zal binárisan kódolt decimális (BCD) formában ábrázolhatók-tárolhatók számok, és műveletek is így végezhetők el.

Milyen is egy decimális szám BCD alakban? A tizes számrendszerű szám egy helyi értékének ábrázolásához négy bit bőven elegendő, így nyolc biten (egy bájton) két decimális számjegyet ábrázolhatunk, úgy, hogy az egyik — az alacsonyabb helyi értékű — számjegyet binárisan a 0—3. biten, a másik, magasabb helyi értékű számjegyet pedig a 4—7. biten tároljuk. Azaz 29 DEC=0010 1001 BCD. Igaz, ez a szám a szokványos módon kiolvasva — hiszen PEEK-kel kiolvashatjuk — egészen más értéket reprezentál. Ezért kell tudni, hogy a kiolvasott érték BCD-ben van!

Ez a számábrázolás és -tárolás kissé pazarló, hiszen öt bájton tizjegyű decimális szám ábrázolható, míg tisztán binárisan vagy hexadecimálisan ugyanezen az öt bájton 2749×10^{14} értékű a maximálisan ábrázolható szám. Ez a pazarlás a műveletek elvégzésekor időben megtérül. S hogy valóban pazarolnánk a memóriahelyet? Ugyan, hol van már az egy kilobájtos ZX81-esek hőskora!

Most pedig nézzük a számábrázolást. Mint minden számítógépnél, itt is eltérés van az egész típusú és a valós típusú (egyszeres pontosságú, lebegőpontos) számok ábrázolása között.

A HT például két bájton tárolja az egész típusú, és négy bájton az egyszeres pontosságú változókat.

Az Enterprise mindkét esetben hat bájtot használ fel a szám tárolására. A hatodik bájt a típus jelzésére alkalmas, valamint arra, hogy az öt bájton normál alakban tárolt szám kitevőjét tárolja. (Persze azért itt is van némi „ravaszság"!)

Tekintsük át részletesen a számábrázolást. Írjuk be az *1. listán* látható programot. A 1Ø-es sorban tetszőleges numerikus változóknak adhatunk értéket, a 2Ø-as sor a változótábla elejét határozza meg, a ciklus pedig rendre kiírja az egyes memóriarekeszek tartalmát.

Igen ám, de mint már szó volt róla, ez így nem ad helyes eredményt. Gondoskodni kell a cikluson belül egy olyan rutinról, amely a memóriarekesz tartalmát bináris vagy hexadecimális formában írja ki, mert így már azonnal felismerhetjük az egy bájton tárolt két decimális helyi érték számjegyeit *(2. lista)*.

Gondolom, a többszöri futtatás során sikerült negatív, illetve egész számokat is a 1Ø-es sorban levő változónak értékül adni. A negatív szám jelzésére a hatványkitevőt tároló bájt hetedik bitje utal. Ha ennek értéke 0, akkor pozitív a szám, ha pedig értéke 1, akkor a szám negatív.

Az egész típusú változók ábrázolásához — mint láttuk a HT példáján — elegendő két bájt, igaz, ilyenkor a számot nem BCD alakban kell ábrázolni. Az első program segíthet az egész típusú változók tárolásának megfigyelésében, megértésében. Igaz viszont, hogy itt nem a két bájton maximálisan tárolható 32758-at, hanem csak 9999-ig tárolja az Enterprise a számokat az N=L+256 H alakban. (Láttuk már ezt a számábrázolási módot és korlátot a programsorok tárolásának vizsgálatánál.)

Ezek ismeretében kukkantsunk be a számítógép ROM-jába. A működéshez a gép egy sor konstanst tárol. Ezek közül jó néhány értékét közvetlenül megkérdezhetjük BASIC kulcsszó segítségével. Például: PRINT PI, PRINT VERNUM, PRINT INF.

De honnan tudja ezeket az értékeket a számítógép? Már láttuk, hogy a számítógép 5+1 bájton tárolja a változó értékét. Van programunk, aminek segítségével ki tudjuk olvasni az adott címtől kezdődő értékeket, így már csak az kellene, hogy ismerjük a memóriaterületek kezdőcímét.

A harmadik program sorban kiírja a számjegyeket és a hatványkitevő (előjel) tárolására szolgáló bájt tartalmát *(3. lista)*.

A 4. lista programja elvégzi a kiolvasást és végeredményként (az egyes bájtok tartalmának kiírása nélkül) kiírja a tárolt szám értékét.

A megfelelő C érték beírásával különböző konstansok értékét kapjuk meg. Lássunk néhány címet, melyeket a 3. és 4. lista 2Ø-as sorába kell a C változó értékeként beírni:

Cím	Konstans
216	2/PI
181	180/PI
160	max. ábrázolható szám
377	LN 10
3533	VERNUM

SZ. LUKÁCS JÁNOS

```
10 LET A=3.14
20 LET M=PEEK(566)+256*PEEK(567)
30 !M=valtozotabla eleje
40 LET S=PEEK(M+3)
50 !S=a vizsgalt valtozonev hossza
60 FOR I=M TO M+S+9
70   PRINT I;PEEK(I)
80 NEXT I
```

1. lista

```
10 LET SAM=1.23456789
20 LET M=PEEK(566)+256*PEEK(567)
30 !   M=valtozo tabla eleje
40 LET S=PEEK(M+3)
50 !   S=valtozo nev hossza
60 FOR I=M TO M+3
70   PRINT I;PEEK(I)
80 NEXT
90 FOR I=I TO I+S-1
100   PRINT I;PEEK(I),CHR$(PEEK(I))
110   ! a valtozonev karakterei
120 NEXT
130 FOR I=I TO I+4
140   LET T=PEEK(I)
150   LET E=INT(T/16)
160   LET M=MOD(T,16)
170   PRINT I;E;M
180   ! a tarolt BCD ertekek
190 NEXT
200 PRINT I;PEEK(I)
210 ! peek(i) itt a hatvanykitevo+64
```

2. lista

```
10 LET C=167 ! ez most 2^15
20 LET C=16384+C
30 OUT 177,1
40 PRINT C;PEEK(C)
50 !       peek(c) = kitevo
60 FOR I=C-1 TO C-5 STEP-1
70   LET T=PEEK(I)
80   LET E=INT(T/16)
90   LET M=MOD(T,16)
100   PRINT I;E;M
110   ! a memoria tartalom BCD-ben
120 NEXT I
130 FOR I=C-1 TO C-5 STEP-1
140   LET T=PEEK(I)
150   PRINT INT(T/16);T BAND 15;
160 NEXT I
170 PRINT
```

3. lista

```
10 LET C=188 ! ez most PI/180
20 LET C=16384+C
30 OUT 177,1
40 LET S=0
50 LET O=1
60 FOR I=C-1 TO C-5 STEP-1
70   LET T=PEEK(I)
80   LET E=INT(T/16)
90   LET M=MOD(T,16)
100   LET S=S+(E+M/10)/O
110   LET O=100*O
120 NEXT
130 LET J=PEEK(C)
140 LET L=(J BAND 128)/128
150 LET J=J-64-128*L
160 LET E=(L=1)-(L=0)
170 PRINT "a=";E*S*10^J
```

4. lista

EXOS 2.1
PÉLDAPROGRAMOK

A Magazin előző számában megismerhettük a legalapvetőbb EXOS-hívásokat. Most lássunk a *listán* egy példaprogramot, amely az ASMON-nal vagy SIMON-nal lefordítva bemutatja az EXOS hibakódjait és egyúttal példa a csatorna-kezelésre is.

ERDEI ANDRÁS

```
;
;       INTRODUCE EXOS ERRORS
;
;         (C) 1988 C.C.Guys
;       written by Erdei András
;
        ORG     0800H
;
        EQU-s
;
    ; EXOS hivasok kodjai
        ; hibakod szoveg    konvertalasa
ASKERR  EQU     20
        ; csatorna nyitas
OPEN    EQU     1
        ; csatorna lezaras
CLOSE   EQU     3
        ; blokk iras
WRBLCK  EQU     8
        ; karakter iras
WRBYTE  EQU     7
        ; specialis funkcio
SPECFN  EQU     11
        ; kep megjelenites
DISP    EQU     1
        ; valtozo olvasas/iras/atkapcs.
EXVAR   EQU     16
    ; renszervaltozok
        ; video csatorna szama
VID_CH  EQU     10
        ; video mod
MODVID  EQU     22
        ; video szinmod
COLVID  EQU     23
        ; kepernyo x-iranyu kiterjedese
X_SIZE  EQU     24
        ; kepernyo y-iranyu kiterjedese
Y_SIZE  EQU     25

SPC     EQU     32      ; szokoz
CR      EQU     13      ; kocsi-vissza
LF      EQU     10      ; soremeles
;
;       MACROS
;
    ; EXOS változó irása
WRVAR   MACRO   Á1
        LD      C,Á1
        LD      B,1
        EXOS    EXVAR
        ENDM
    ; karakter kiirása a kepernyore
PRINT   MACRO   Á1
        LD      A,VID_CH
        LD      B,Á1
        EXOS    WRBYTE
        ENDM
;
;       CODE
;
        ; EXOS hibak bemutatasa
ERRORS
        ; video opciok beallitasa
        LD      D,0
        WRVAR   MODVID
        LD      D,0
        WRVAR   COLVID
        LD      D,20
        WRVAR   Y_SIZE
        LD      D,40
        WRVAR   X_SIZE
        ; video csatorna megnyitasa
        LD      A,VID_CH
        LD      DE,VIDNAM
        EXOS    OPEN
se
        LD      A,VID_CH
        LD      B,DISP
        LD      C,B
        LD      E,B
        LD      D,20
        EXOS    SPECFN
        ; hibakod
        LD      B,0
NXTERR
        PUSH    BC
        PUSH    BC
        ; hiba szovege a pufferbe
        LD      A,B
        LD      DE,ERRTXT
        EXOS    ASKERR
        POP     AF
        ; hibakod kiirasa
        CALL    WRHEXB
        PRINT   SPC
        ; hiba szovegenek kiirasa
        LD      DE,ERRTXT
        LD      A,(DE)
        INC     DE
        LD      C,A
        LD      B,0
        LD      A,VID_CH
        EXOS    WRBLCK
        PRINT   CR
        PRINT   LF
        ; varakozas
        ;       a kiiras a PAUSE
        ;       gombbal is megallithato
        LD      BC,0
WAIT    DEC     BC
        LD      A,B
        OR      C
        JR      NZ,WAIT
        ; kovetkezo hiba
        POP     BC
        DJNZ    NXTERR
```

```
p megsemmisitese
        LD      A,VID_CH
        EXOS    CLOSE
        RET
;
    ; byte hexadecimalis kiirasa
WRHEXB
        PUSH    AF
        RRCA
        RRCA
        RRCA
        RRCA
        CALL    WRTETR
        POP     AF
WRTETR
        AND     0FH
        ADD     A,90H
        DAA
        ADC     A,40H
        DAA
        LD      B,A
        LD      A,VID_CH
        EXOS    WRBYTE
        RET
;
;       DATAS
;
        ; hely a hibauzenet szamara
ERRTXT  DEFS    64
        ; video egyseg neve
VIDNAM  DEFB    6
        DEFM    "VIDEO:"
;
        END
;
```

```
  1 ! KOSZEGI TAMAS
 100 GRAPHICS HIRES 4
 110 INPUT PROMPT "SZELESSEG.HOSSZUSAG.MA
GASSAG":X,Y,Z
 120 LET W=300:LET 0=SQR(3)
 130 LET XA1,XA2=2*W-0*X/2
 140 LET YA1=W-X/2
 150 LET YA2=2-X/2+W
 160 LET XA3,XA4=0*(Y-X)/2+2*W
 170 LET YA3=2-(X+Y)/2+W
 180 LET YA4=W-(X+Y)/Z
 190 LET XB1,XB2=2*W
 200 LET YB1=W
 210 LET YB2=Z+W
 220 LET XB3,XB4=0*Y/2+2*W
 230 LET YB3=2-Y/2+W
 240 LET YB4=W-Y/2
 250 SET INK 255
 260 PLOT XA1,YA1;XA2,YA2;XA3,YA3;XA4,YA4
;XA1,YA1
 270 PLOT XB1,YB1;XB2,YB2;XB3,YB3;XB4,YB4
;XB1,YB1
 280 PLOT XA1,YA1;XB1,YB1
 290 PLOT XA2,YA2;XB2,YB2
 300 PLOT XA3,YA3;XB3,YB3
 310 PLOT XA4,YA4;XB4,YB4
 320 END
```

TÉGLATEST A KÉPERNYŐN

Az IS-BASIC lehetőséget nyújt arra, hogy a geometriai formákat egyszerű eszközökkel megjelenítsük. Igaz, meg kell alkudni a képernyőn látható grafikus lehetőséggel és a programlista áttekinthetőségével. (Mindezzel olvasóink a következő számunkban részletesen megismerkedhetnek.)

A közölt program egy téglatestet három dimenzióban tesz láthatóvá. Természetesen ez egy térbeli idomtól, illetve az azt megjelenítő programtól elvárható, de nem mindegy, hogy az ilyen ábrázolást milyen programozási módszerrel érhetjük el. Az IS-BASIC egyik nagy előnye, hogy ennek megoldása viszonylag egyszerű.

A start után a program megkérdezi a téglatest hosszúságát, magasságát és szélességét. Így előáll a képernyőn a háromdimenziós kép. Az Enterprise-tulajdonosokra bízzuk, hogy azt körbe is forgassák. Ha nem sikerül, legközelebb megnézhetik a megoldást. Ügyeljünk a téglatest méreteire, hogy a képernyőn maradhassunk.

KŐSZEGI TAMÁS

Mi a manó?

Fölös kontroller továbbra is. A Centrum és a Videoton nem tudott megegyezni abban, hogyan bontsák meg a lemezmeghajtó egységcsomagját. Amint az előző számunkban írtuk, az az Enterprise-tulajdonos, aki Videoton lemezmeghajtót vásárol, kénytelen gazdagodni egy TVC kontrollerrel és szegényedni pár ezer forinttal. A Centrum a sikertelen tárgyalás ellenére is három budapesti márkaboltjában szétválasztja a kényszerű árukapcsolást.

■

Számítógép OTP-hitelre. Magyarországon először kapható számítógép OTP-hitelre a Centrum új akciója révén. Így 3900 forint előleg befizetésével azok a gépkedvelők is hozzájuthatnak az Enterprise-hoz, akiknek az amúgy nem magas gépár is megterhelő lenne.

■

Enterprise közületnek. A profi felhasználók is jól alkalmazhatják az Enterprise-t kisegítő gépként. Emellett szólnak az alábbi lehetőségek:

— Harmincként gép átalakítás nélkül hálózatba köthető.

— A gép MS-DOS adatkompatibilis és Commodore lemezt is tud olvasni. Így azok a cégek, amelyek C64-ről akarnak IBM kompatibilis gépre átállni, a meglévő adataikat az Enterprise segítségével megmenthetik.

— CP/M operációs rendszer alatt futó programok alkalmasak egyéni feladatok megoldására, amik aztán átvihetők nagygépre is.

— A PS/2-re átálló cégek a különböző méretű lemezeiket konvertálhatják az Enterprisezal.

A közületi alkalmazás nem megy a lakossági ellátás rovására, mert gép van elég. A programfejlesztés azonban valószínűleg fellendül.

■

Hardverfilozófia. Az Enterprise gépekhez árusított hardverkiegészítők ára a Centrumnál eleve tartalmazza a géphez illesztés költségét. A vásárlási blokk bemutatása után a szerviz díjtalanul elvégzi az átalakítást.

■

Programkazetta-kölcsönzés. Hasonlóan a videokazettához, várhatóan programkazetták is kölcsönözhetők lesznek. A Centrum és az „A" Stúdió karácsonyra tervezi az új szolgáltatás megkezdését. Akinek a kazettán lévő szoftver megtetszik, kedvezményes áron meg is vásárolhatja.

MEGKÉRDEZTÜK AZ ENTERPRISE -RÓL

Ezúttal Kopácsy Vilmostól kértünk interjút, aki az „A" Stúdióban a nyolcbites számítógépek rendszerfelelőse. Az „A" Stúdió magja 1984-ben jött létre, majd szakcsoportként működtek, míg végül kisszövetkezetté nőtték ki magukat. Szakmai körök általános véleménye szerint amit az Enterprise-ról tudni lehet, azt ők tudják. A Centrummal való együttműködés alapján szoftvereket fejlesztenek és könyveket írnak az Enterprise-hoz.

M. M. Mikor és hogyan kerültek először kapcsolatba az Enterprise-zal?

K. V. Amikor az Enterprise 1985-ben megjelent, a cégnek — hasonlóan a későbbi magyar helyzethez — a gép bevezetéséhez programokra volt szüksége. Ezért megkeresett szakmailag párhuzamos vállalatokat, olyanokat, mint az Entersoft szoftverkiadót és más cégeket. Így teremtettek kapcsolatot a Novotrade-del is. A Novotrade az „A" Stúdióval kötött programfejlesztésre szerződést. Nehéz körülmények között, hiányos dokumentációkra támaszkodva, két játék- és egy sprite-tervező programot készítettünk. Ez jó alkalom volt a Magyarországon még ismeretlen Enterprise tanulmányozására.

M. M. Mi volt a vélemény a munkájukról?

K. V. Szakmailag elégedettek voltak, igaz, közben az Enterprise cég pénzügyi helyzete megingott.

M. M. Hogyan jutott eszükbe a Spectrum-emulátor fejlesztése?

K. V. Az angolok arra törekedtek, hogy minél több szoftver legyen a géphez. Mi felismertük a két gép közötti hasonlóságot, és egyre kézenfekvőbb lett a megoldás. Az Enterprise cég bizonytalan anyagi helyzete miatt a tervek fiókba kerültek. Nem a teljes cég ment tönkre, a német része fennmaradt, de ez nem sokat lendített az ügyeken.

M. M. A szerződésük lejárta után miként elevenítették fel ismét a kapcsolatot az Enterprise-zal?

K. V. Mi egy pillanatra sem felejtettük el ezt a gépet. Igaz, hogy a dokumentumok a fiókban pihentek, de az Enterprise fejlesztőgéppé nőtte ki magát az „A" Stúdióban.

M. M. Milyen fejlesztéseik voltak?

K. V. Mindenekelőtt megrendelésre meglévő szoftvereket átírtunk másik gépre. Többek között a C64-es programokat Spectrumra. Ebben az esetben azon a gépen fejlesztettünk, amelyiken a program fut, nagyon körülményes és lassú. Ezért a teljes programrendszert a fordítással együtt az Enterprise-ra vittük, ami összeköttetésben volt a célgéppel. A fejlesztési műveleteket az Enterprise-on végeztük el, és az eredményt átadtuk annak a gépnek, amelyen futnia kell. Így nemcsak időt lehet nyerni, hanem felesleges munka is megtakarítható, mert ha a program „elszáll", még mindig rajta marad a főgépen. Az Enterprise-t ideálisan lehet használni cross-fejlesztésre.

M. M. Mit szóltak hozzá, amikor megkezdődött az Enterprise gépek importja?

K. V. Nagyon jó ötletnek tartottuk, különösen a gép tudását és minőségét figyelembe véve. Messzemenően jobb, mint az akkori köztudatban domináló C64. Az a véleményem, kinőttünk már abból, hogy csak játszani akarunk. Az emberek lassacskán otthon is komolyabb feladatokat akarnak megoldani.

M. M. Az importot követően megújult az üzleti kapcsolatuk az Enterprise céggel?

K. V. Velük nem. Tény azonban, hogy Dániában, Hollandiában és Angliában képviselve vagyunk. Ezekben az országokban is forgalmazták az Enterprise-t, és ennek a szupportjához járulunk hozzá. Nyugaton is jó pár ezer ilyen gép kelt el, és kellenek a programok. Az ottani forgalmazókkal is nagyon jó a kapcsolatunk, programokat fejlesztünk számukra, és adott esetben értékesítési jogokat cserélünk.

M. M. Önök voltak egy ideig a legilletékesebbek a Spectrum-emulátor gyártásában. Mi az igazság a nagy port kavart ügyben?

K. V. Ahogy közhelyként mondani szokták: Magyarországon élünk, és ez a körülmény behatárolja a munkánkat is. Az emulátort megpróbáltuk gyártani, és ezzel betonfalnak rohantunk! Többek között a tervezés előtt megkérdeztünk minden erre hivatott vállalatot, hogy milyen alkatrészek kaphatók. Mire a gyártás megkezdődött volna, a tervezett alkatrész eltűnt a piacról. Ezután jött a doboz! Amelyik dobozt elfogadható áron gyártották volna, az minőségileg és esztétikailag is alkalmatlan volt. Ezután merült fel az a gondolat, hogy az emulátort a Videoton Automatika Leányvállalat gyártsa. Ez mindenkinek jobb volt, de úgy tudom, ők is nehézségekkel küzdenek.

M. M. Nem látja sötétnek az Enterprise jelenlegi helyzetét, illetve a jövőjét?

K. V. Ha ilyennek látnám, akkor nem lenne egy Enterprise az asztalomon egy PC/AT és Atari ST társaságában.

M. M. Melyek az „A" Stúdió ígéretes újdonságai a hazai Enterprise-tulajdonosok számára?

K. V. Jelenleg hat szoftveren dolgozunk. Munkánkat alapvetően két részre bontom: egyrészről a szórakoztatásra, másrészről a komolyabb programozást célzó fejlesztésekre. Ilyen a BASIC-fordító és a BASIC-bővítés, amely a gépi kód és a BASIC-utasítás optimális arányát teremti meg, ráadásul még hatvan függvényt is tartalmaz. Szándékunk hosszú távon foglalkozni az Enterprise-zal, természetesen ameddig üzletileg is megéri.

M. M. A programlopásokra gondol?

K. V. Nemcsak arra, habár igaz, hogy ez is rontja üzleti érdekeinket. A szoftverfejlesztés az egyik legrizikósabb vállalkozás. Egy középszintű játékprogram kifejlesztése négy ember hat hónapig tartó munkájának eredménye. Az értékesített programok után darabonként tizenöt forint szerzői rész jut átlagosan. Ebből könnyen ki lehet számítani a programozó félévi jövedelmét. Ugyanakkor a fejlesztés nagyon eszközigényes, nagy a rezsije és drága számítógépek kellenek hozzá.

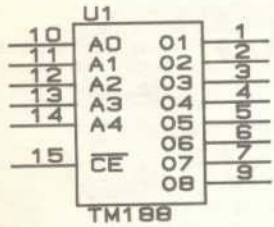
M. M. Azt hallottam, hogy már a fejlesztéskor is kijutnak kalózpéldányok. Ez hogyan történhet meg?

K. V. Tőlünk biztosan nem. A tesztelőkön és a gyártás során kerülhetnek ki, ami megmagyarázhatatlan magatartás a tesztelők és a gyártók részéről.

PINKE GYÖRGY

Hardver

A sorozat alapgondolata — azon a régi felismerésen túl, hogy az elektronika és a számítástechnika elválaszthatatlan egymástól — a következő tapasztalatot summázza. A szoftver — a programok — jelentősége egyre nő, de az is tény, hogy az igazán jó (az adott számítógép nyújtotta lehetőségeket maximálisan kihasználó) programok megírásához a programozónak rendelkeznie kell alapfokú áramköri hardverismerettel is. Megerősíti ezt, hogy szaporodik az olyan berendezések, mikroprocesszort alkalmazó rendszerek száma, amelyek programvezérelten működnek. Az ilyen rendszerek tervezőinek és fejlesztőinek is szükségük van integrált hardver- és szoftverismeretekre.



1. ábra

Az EPROM-okról szóló ismertetés után még három olyan, csak olvasható memóriatípust mutatunk be, amellyel a számítógépekben találkozhatunk.

PROM

A felhasználó által egyszer programozható memória. Ennél a típusnál minden, a tárolóban lévő bitnél egy fémszálból kialakított, a programozás során egy nagyobb árammal kiégethető „olvadó biztosító" található. Egy programozókészülék segítségével megcimezzük a PROM egyes szavait. Ha a szó azon bitjeihez kapcsolódó fémszálakon, amelyeket meg kell változtatni, nagyobb áramot küldünk át, az elolvasztja a fémszálat és az elektromos vezetés megszűnik. Akárcsak az EPROM programozók, ezek is képesek automatikusan működni és önállóan programozni egy teljes tárterületet. Az *1. ábrán* egy jellegzetes, gyakran használt IC, a 32×8 bit kapacitású TM188-as bekötése látható.

MASZKOLT ROM

A memória tartalmát a gyártó a megrendelő kívánsága szerint fixen, ún. maszkolással (a chipre ráfotózva) írja be. Ez a maszk tulajdonképpen a beírandó tártartalom geometriai képe. Ilyen maszkolt ROM például a C64-es Kernal ROM-ja.

ELEKTROMOSAN TÖRÖLHETŐ ROM

Az EPROM-ok továbbfejlesztésének tekinthető. Programozása az EPROM-okéhoz hasonló, törlése azonban nem ultraibolya fény, hanem elektromos impulzus hatására megy végbe. Felhasználási köre egyre bővül.

ÍRHATÓ-OLVASHATÓ MEMÓRIÁK: A RAM-OK

A mikroprocesszoros rendszerekben az információk átmeneti tárolására írható-olvasható memóriákat alkalmaznak. Ezekben kétállapotú tárolóeszköz van: a statikus RAM-okban a már bemutatott flip-flop áramkör, a dinamikus RAM-okban pedig egy tranzisztorral mint kapcsolóval töltött és kisütött kondenzátor.

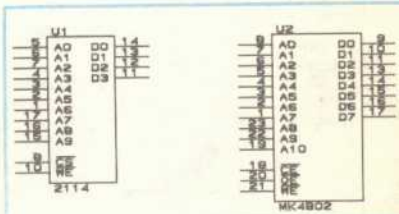
STATIKUS RAM

Az ilyen memóriákban a tárkapacitással megegyező számú flip-flop van. A tárolók beírás után tartalmukat a tápfeszültség kikapcsolásáig megőrzik. Erre utal az elnevezésükben szereplő „statikus" kifejezés. Ezeknek a RAM-oknak a jelvezetékei azonosak a már tárgyalt EPROM-okéival: címzővonalak, adatvonalak, tokkiválasztó jelek, de ezeken kívül itt szükség van a két alapvető működési módot, az írást és olvasást biztosító jelre is. Ez vagy egy összevont vezérlőjel, az R/W, vagy két önálló jel: a WR=írás és az RD=olvasás. A *2. ábrán* a két talán leggyakrabban használt típust, a 2114 és az MK4802 áramkört mutatjuk be.

A 2114 áramkör 1024×4 bit szervezésű. Egyetlen +5 V-os tápfeszültséget igényel, valamennyi bemenete és kimenete TTL kompatibilis.

Az MK 4802 áramkör bit szervezésű. Az átgondolt lábkiosztás eredményeként ez a tok szinte változtatás nélkül egy 2716 típusú EPROM helyére dugható, feltéve, hogy a 21-es lába a WR jelet kapja meg. Ezzel a megoldással, az EPROM foglalatba RAM-ot dugva, igen egyszerűen fejleszthetjük, módosíthatjuk a később véglegesen az EPROM-ba kerülő programot.

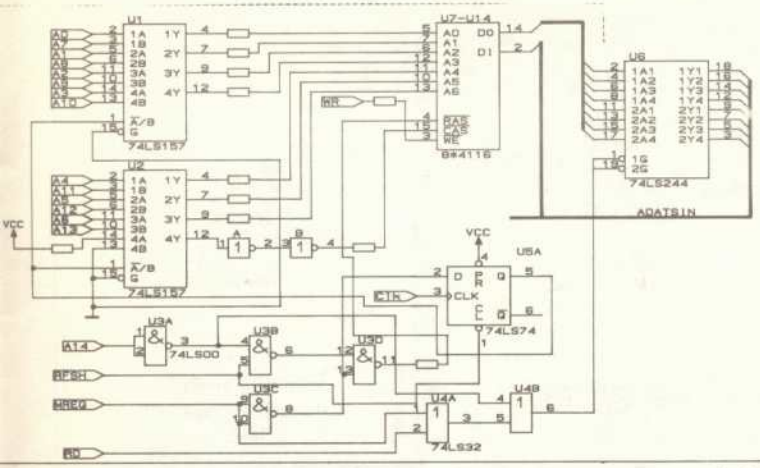
Jogos igény, hogy a tápfeszültség kikapcsolása után a RAM memória megőrizze a tartalmát. Ezt a kis fogyasztású, CMOS technológiával készült RAM áramkörök teszik lehetővé. Ezek a RAM-ok funkcionálisan teljes mértékben megegyeznek a hagyományosakkal. Jellegzetes, a gyakorlatban széles körben használatos a Hitachi cég HM6116-os két kbájtos és HM6164-es nyolc kbájtos típusa. Jelenleg már kaphatók 32 kbájtosak is. Bekötésük megegyezik a statikus RAM-okéval. Ahhoz, hogy beírt tartalmukat megőrizzék, nem elég, hogy kikapcsolás után külön telepről kapják a tápfeszültségeket. Ugyanis mikor egy mikroprocesszoros rendszert kikapcsolunk, a rendszer „összeomlásakor" kialakulhat a tokra kapcsolódó jelek olyan kombinációja, hogy a tok véletlenszerűen kiválasztódik, és valamilyen információ beíródik az egyik vagy másik rekeszbe. Ezt úgy akadályozhatjuk meg, hogy a kikapcsolás után első pillanattól kezdve, mikor a tápfeszültség még csak kismértékben csökken, gondoskodunk arról, hogy a tokot kiválasztó jel, a CS ne kerülhessen aktív (általában alacsony, L szintű) állapotba. Erre példa a *3. ábrán* bemutatott, szaklapból vett megoldás. Míg a rendszer tápfeszültsége megvan, addig a Q1 tranzisztoron keresztül a tokot kiválasztó CS jel változatlanul átjut. A tápfeszültség kikapcsolása után a Q1 tranzisztor a Q2 segítségével lezár (Q1 is és Q2 is zárva), ami megakadályozza, hogy a CS kivezetésre alacsony szintű impulzus jusson. A 3,6 V-os nikkel-kadmium akkumulátor biztosítja a kikapcsolt állapotban a RAM igen kis tápáramát és a CS jel magas szintjét. Amikor a tápfeszültség megvan, az akkumulátor a D4—R13 útvonalon keresztül töltődhet.



2. ábra

DINAMIKUS RAM

A dinamikus RAM-okban az információt nem flip-flop, hanem egy térvezérlésű (FET) tranzisztor belső kapacitása mint kondenzátor tárolja, az itt nem részletezett módon. A feltöltött kondenzátor felel meg az egyik logikai állapotnak, és a kisütött, azaz töltés nélküli állapot képviseli a másikat. Sajnos ez a töltés egy idő múlva a szigetelőrétegen keresztül eltűnik, azaz önmagától is kisül a kondenzátor. A gyártók általában 2 ms-on belül írják elő az ilyen módon felírt dinamikus RAM tartalmának megújítását, azaz a frissítést. A frissítés (angol szóval refresh, ejtsd rifres) során a kisülni kezdő kapacitásokat ismét feltöltik. Ezt a műveletet a dinamikus tárban lévő



4. ábra

áramkör automatikusan végzi, csupán a frissíteni kívánt tárrész ún. frissítési címét kell megadni.

A dinamikus RAM-ok általában speciális társzervezésű, egybites tárak, a tár négyzetes alakban, mátrixként van szervezve. Minden bit egy elképzelt négyzetháló metszéspontjaiban helyezkedik el, azaz egy bit helyét a háló sorának és oszlopának értékével azonosíthatjuk. A frissítést megkönnyíti, hogy csak a sorcímet kell kiküldeni, és ennek hatására a sorcímhez tartozó teljes oszlop frissítése megtörténik. Ez azt jelenti, hogy például egy 16 kbites, 128 × 128-as négyzetbe szervezett tár mindössze 128 lépésben frissíthető.

Mivel a tároláshoz csupán egy FET tranzisztorra van szükség bitenként, ezért igen könnyű nagy kapacitású dinamikus tárakat megvalósítani. A jelenleg csaknem minden személyi számítógépben használt típusok: a 4116-os 16 kbites, a 4164-es 64 kbites és a 41256-os 256 kbites dinamikus tár. Az ilyen áramköröket felhasználó, bájt szervezésű tároknál 8 tokkal valósítható meg a nyolcbites adatszélesség.

A tokkivezetések számának csökkentése érdekében, a mátrixos szervezésből adódóan megtehetjük, hogy közös címvezetékkel címezzük meg a négyzetet (mátrixot) alkotó sorokat és oszlopokat. Azért, hogy a közös lábon, de időben eltolva megjelenő sor- és oszlopcímeket meg tudjuk különböztetni, külön jelek azonosítják azt, hogy éppen mi található a címbemeneteken.

A 64 kbites 4164-es típus a 16 bites címet két nyolcbites darabban fogadja. Az először beérkező nyolcbites címrész a tokon belül sorcímként értelmeződik. A beíró jelét a RAS kivezetésre kell kapcsolni, a felső nyolc bit lesz az oszlopcím, aminek érvényességét a CAS kivezetésre kapcsolt jel jelzi. Egy ilyen memória megfelelő működésének két fontos feltétele van:

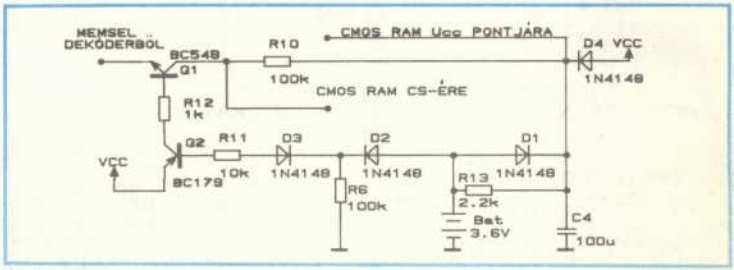
— a mikroprocesszor által kiadott 16 bites memóriacímnek két részre osztva, időben eltolva kell megjelennie a dinamikus RAM címbemenetein;

— elő kell állítani ezen sor- és oszlopcímek érvényességét jelző RAS és CAS jeleket.

Most nézzük meg egy konkrét, a *4. ábrán* látható kapcsoláson, hogyan is működik egy 16 kbájtos, Z80 mikroprocesszorhoz kapcsolt dinamikus tár. Az U1 és U2 jelű multiplexerek a 14 bites címet 2 × 7 részre bontják. A multiplexerek vezérlését és a dinamikus RAM-ok vezérlőjeleinek előállítását a NÉS kapukból, inverterekből és D tárolóból kialakított hálózat végzi a Z80 mikroprocesszor vezérlőjeleinek felhasználásával.

Összefoglalva az eddigieket: a memóriák a mikrogép legfontosabb kiegészítő elemei. A mikrogép működtető programjainak tárolására EPROM memóriák szolgálnak. Kisebb rendszerekben, ahol nincs szükség nagy RAM memóriára, csaknem kizárólag statikus RAM-okat alkalmaznak. Személyi számítógépekben, ahol a programok és az adatok tárolása nagy RAM-területeket igényel, olcsóságuk és nagy tárolókapacitásuk miatt kizárólag dinamikus tárakat használnak.

DR. KÓNYA LÁSZLÓ



3. ábra

Vállalkozási és Kereskedelmi Kft.
1112 Budapest, Budaörsi út 42.
Telefon: 666-617 Telex: 22-7676 md h

Az MD Kft. legújabb ajánlata!
Működésképtelen Winchesterét átvesszük és kedvező áron újat adunk!

— **SEAGATE**	ST 225	25,6 Mb	65 ms	35 000 Ft + ÁFA
	ST 238	38,4 Mb	65 ms	39 000 Ft + ÁFA
	ST 251	51,2 Mb	40 ms	69 000 Ft + ÁFA
	ST 251–1	51,2 Mb	30 ms	79 000 Ft + ÁFA
	ST 4096	85 Mb	30 ms	129 000 Ft + ÁFA

— **MD—GUTOR** szünetmentes áramforrás a legfejlettebb technológia képviselője
teljesítmény: 0,5—10 kVA

— **IBM** felújított perifériák nagy választéka
IBM 3320 lemezegység
IBM 3420 mágnesszalag
IBM 3203 rendszernyomtató stb.
Szállítás raktárról! 1 év garancia!

BÖRZE

TUTTI

ELECTROCOOP KISSZÖVETKEZET

- IBM PC kompatibilis gépek
- HARDVERTELEPÍTÉS SZERVIZ ÉS GARANCIA
- SZOFTVERES TÁMOGATÁS
- RÖVID HATÁRIDŐ

Cím: Bp., Üllői út 81. 1091
Tel.: 334-354

ECOSOFT
Számítástechnikai
Szolgáltató
Kisszövetkezet

IBM PC/AT
kompatibilis számítógép
ár: 239 000 Ft-tól
TURBO/32
32 bit, 24 MHz
ár: 479 000 Ft-tól.
Az általunk forgalmazott eszközök lizingelhetők is.
Tel.: 863-677

---

tanácsadás, szoftvertermékek,
oktatás, papíralapú és
hardver-, mágneses adathordozók,
szoftverbemutatók, számítástechnikai
programkészítés, segédeszközök,
helyi gépbérlet szakirodalom

COMPUTER—M
MINDEN MEGYÉBEN!

Hardvereszközök beszerzése, értékesítése.

MIKROPO BUDAPEST **IRODA:**
VI., Nagymező u. 51.
TEL.: 325-768

- **EPROMÉGETÉS**
  2716—27512-ig bármely típus, AZONNAL IS!!!
- **ADATRÖGZÍTÉST**
  vállalunk d BASE formátumban floppylemezre!
- **konvertálási lehetőség**

---

asy **Telefon: 415-166**
Kereskedelmi és szoftveriroda
1061 Bp., Liszt F. tér 10.
Telex: 22-4378

- **ASY—16 szupermikro számítógép** — 12 terminál
  — VME busz
  — UNIX
- **CRT TERMINÁLOK**
  VT—52, QUT—102, Siemens 8160
- **BILLENTYŰZETEK**
- **MONITOROK**
- **INTEGRÁLT VÁLLALATI INFORMÁCIÓS RENDSZER**

UNIX környezetben üzemeltethető.

PERIFÉRIA
Elektronikai
Fejlesztő
és Szolgáltató
Kisszövetkezet
Bp. VII., Peterdy u. 30.
Telefon: 213-588

ajánlata:

- **P—XT:** 140 E Ft-tól + áfa
- **P—AT:** 200 E Ft-tól + áfa
- igény szerinti konfigurációk
- **FX—1000** PRINTER 90 E Ft + áfa

---

procontrol **ez a védjegy a megbízható**

termékeket jelöli

- Profi XT, AT, 386 gépek
- PLATON folyamatirányítók
- PORTAPRINT blokkolóórák
- T80 biztonsági rendszerek
- BCR vonalkód eszközök

PROCONTROL
KISSZÖVETKEZET
6725 SZEGED,
VERESÁCS U. 28/B
TEL.: 62/21-165, 28-985,
TELEX: 82-726

ERIKA

a legújabb hordozható irodai elektronikus írógép.
CSÖNDES, PONTOS, MEGBÍZHATÓ,
ára: 25 200 Ft,
áfa-val együtt.
VÁSÁROLJA MEG nálunk:
Bp. VI., Népköztársaság útja 2.

ECONORG SZÁMÍTÁSTECHNIKAI KÖZÖS VÁLLALAT

1146 Bp., AJTÓSI DÜRER SOR 10.
Levélcím:
1393 Pf.: 319.
Telefon: 421-974
Telex: 22-6544

november havi kínálata:

a 80 386-os mikroproceszorra alapozott, multiuser üzemmódú

ACER SYS 32

rendszer, amely széleskörűen, változtatható kiépítésben, alap és alkalmazói szoftverekkel együtt kapható.

Az Ön feje nem adattárolásra való, hanem fontos döntések hozatalára.
Joggal várhatja el, hogy kezében legyen az eszköz, ami munkáját minőségivé teszi, döntését megalapozza.
A számítástechnikában viszont a széles választékból nem könnyű a legjobb mellett dönteni.

az Ön fejével gondolkodtunk,

amikor létrehoztuk az első, Közép-Európában egyedülálló számítástechnikai szalont.
Meghívtuk a legfontosabb gyártókat és forgalmazókat, hogy a választék együtt legyen áttekinthető, kipróbálható, tanulmányozható, összehasonlítható.
Felkészült szakembereink várják az érdeklődőket, a leendő vásárlókat.
Reméljük, döntésünk új korszakot nyit az Ön mindennapi munkájában.

PC szalon

Budapest XIII., Sallai Imre u. 6.
315-136, 310-776

Lépjen új korszakba velünk.

MOMTRADE

ISKOLASZÁMÍTÓGÉP-SZERVIZ

Budapest VII., Baross tér 19. 1077
Telefon: 428-909

Vállalja:
IBM PC/AT, IBM PC/XT és Commodore típusú (C16, C Plus/4, C64, C128) gépek javítását, átalánydíjas szervizét, egyedi programok, programcsomagok készítését.

A Helyközi Távbeszélő Igazgatóság Számítóközpontja felvételre keres:

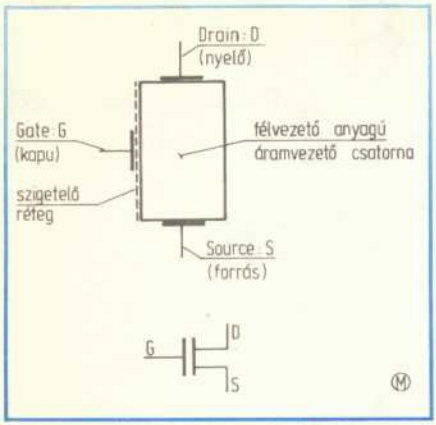
- TPA 1148-as számítógépre gyakorlott rendszerprogramozót, orosz és angol nyelvtudással előnyben,
- érettségizett, gépelni tudó, ügyintézésben jártas adminisztrátort,
- középiskolai végzettséggel rendelkező, TPA 1148 nagygépes, illetve IBM kompatibilis PC-s ismeretekkel bíró futtatási felelőst.

Egyéves munkaviszony után 33%-os vasúti utazási igazolvány, sokféle üdülési kedvezmény, postai kedvezmények. Bérezés kollektív szerződés alapján.

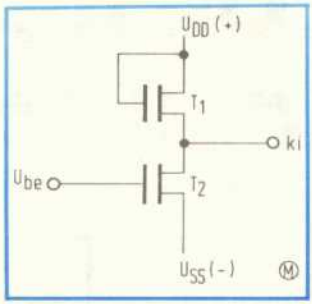
Jelentkezni lehet:

Helyközi Távbeszélő Igazgatóság Számítóközpontja
Budapest VIII., Horváth Mihály tér 17-19.
Telefon: 340-797

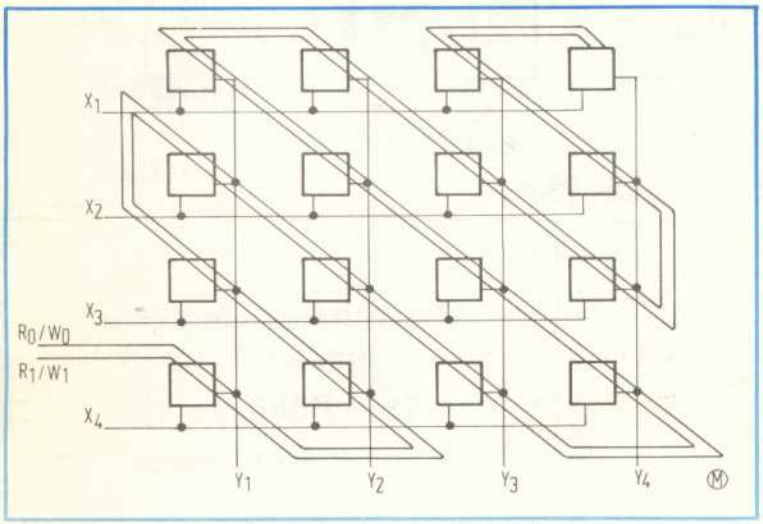
MI VA



1. ábra



2. ábra



3. ábra

Dr. Kónya László Rendszerfejlesztési eszközök című sorozatában már szó volt a tárakról (1988/10. szám). A sorozat cikkei elsősorban felhasználásorientáltak, tehát csak annyira elemzik a fizikai működést, hogy az áramkörök alkalmazását lehetővé tegyék. Úgy gondoltuk, nem érdektelen megismerkedni az egységek belső felépítésével és a bennük lejátszódó fizikai folyamatokkal sem.

A tárak a számítógépek lényeges elemei. Ezekről az egységekről a gépeket alkalmazók, a számítástechnikát már alkotó módon művelők mégis általában csak annyit mondhatnak: a „fekete dobozok". Amikről azt tudják, hogy hány bájt „fér beléjük" (hány rekeszük van), esetleg még azt, hogy mekkora a hozzáférési idejük. Mivel erről a témáról magyar nyelven meglehetősen kevés információ jelent meg, ráadásul többségük túlságosan magas színvonalú, azaz a számítástechnikát művelők többsége számára érthetetlen, úgy gondoltuk, nem haszontalan ezekről az eszközökről írni. És ha már ezt tesszük, úgy csináljuk, hogy az elektronikában kevéssé jártasak is megérthessék! Előismeretként csak némi fizikát kívánunk, az áttekintéshez szükséges elektronikai alapismereteket.

Először röviden ismertetjük a MOS-tranzisztor és a MOS-inverter működését. A MOS-tranzisztoroknak igen sok — a működésben nagy különbségeket mutató — típusuk van, ezért csak egy olyan egyszerűsített leírást adunk, amely a fizikai valóságot csak megközelíti, de azért mindenki megértheti a lényegét. A tranzisztor felépítésének elvi vázlatát és a szimbolikus jelölést az *1. ábrán* láthatjuk.

A működés ismertetéséhez tételezzük fel, hogy a nyelő (drain) a forráshoz (source) képest pozitív feszültségen van. Az áramvezető csatorna technológiai kialakítása olyan, hogy ha a kapu (gate) feszültsége a forráshoz képest negatív (esetleg „csak" nulla), a kapufeszültséggel keltett villamos tér a nyelő—forrás közötti áramúton nem engedi áthaladni a forrásból a nyelő felé mozogni törekvő töltéshordozókat, elektronokat. (A kapufeszültség az áramvezető csatorna keresztmetszetét leszűkíti, elzárja azt.) Pozitív kapufeszültségnél az áramvezető csatorna „megnyílik", és az elektronáram a forrásból akadálytalanul folyik a nyelő felé. Figyelembe kell venni, hogy a kapu és az áramvezető csatorna között szigetelőréteg van, így a kapu áramot nem „fogyaszt", ellentétben a bipoláris tranzisztorokkal, ahol az emitter—kollektor (ez a MOS-nál a forrás—nyelő) körben folyó áram vezérléséhez a bázisnak (ez a MOS-nál a kapu) áramfelvétele is van.

Röviden: a MOS-tranzisztort olyan kapcsolóként foghatjuk fel, ahol a forrás—nyelő kör ellenállását és így a rajta átfolyó nyelőáramot a kapufeszültséggel vezérelni lehet.

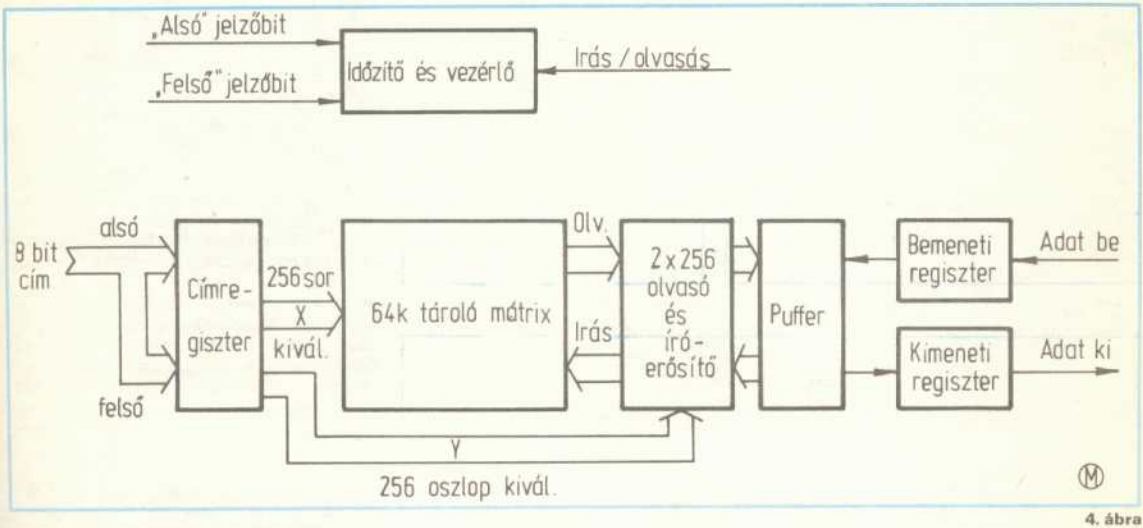
Az elmondottakból következik az is, hogy meghatározott technológiai kivitel (a rétegek geometriai méretei, vezetőképessége stb.) és állandó kapu-forrás feszültségnél (U_{GS}) a MOS nyelő—forrás köre állandó ellenállásként funkcionál, azaz egy MOS-tranzisztor egy ellenállás helyettesítésére is alkalmas. Ennek a ténynek a jelentőségét akkor mérhetjük fel igazán, ha arra gondolunk, hogy egy integrált áramkör „belsejében" egy ellenállás sokkal nagyobb helyet foglal el, mint egy MOS-tranzisztor. Ebből az következik, hogy az integráltság fokának növelésére — igen egyszerű megoldásként — az ellenállások helyettesíthetők MOS-tranzisztorokkal. Erre mutat példát a *2. ábrán* látható MOS-inverterkapcsolás.

Az áramkörben a T_2 a tulajdonképpeni vezérelt eszköz, a T_1 az állandó értékű munkaellenállás. (A szaknyelv a vezérelt tranzisztort meghajtótranzisztornak, a munkaellenállást terhelőtranzisztornak nevezi.) Ha a bemenetre adott feszültséggel — ez lényegében az U_{GS} — a T_2-t vezéreljük, az azon átfolyó nyelőáram erőssége változik, így a T_1-T_2 kör úgy működik, mint egy vezérelhető feszültségosztó. A kapcsolás felépítéséből adódik, hogy növekvő bemeneti feszültséghez csökkenő kimeneti feszültség tartozik, tehát az áramkör invertál.

Társzervezés

Az egy IC-tokban megvalósított tárak

N A CELLÁBAN?



4. ábra

kapacitása a mai technikai színvonalon 4 k—256 k rekesz. Az egyszerűség kedvéért először tételezzük fel, hogy egy rekesz tárolási kapacitása 1 bit (más kifejezéssel: a rekesz egy cellából áll).

A nagy tárolási kapacitás a cellák mátrixba szervezését kívánja meg. Egy 16 bites 4×4-es mátrix felépítését szemlélteti a *3. ábra*. A cellák elrendezése és összekapcsolása olyan, hogy egy-egy X, illetve Y vezeték kijelölése egy cellát választ ki a mátrixban. Minden cellához csatlakozik a két iró-olvasó — R/W — vezeték, amelyeken a cellák tartalma kiolvasható, illetve megváltoztatható. Az elrendezés szabályossága a különböző társszervezési megoldásoknál mindenütt felismerhető, a különbség legfeljebb a vezetékek számában és szerepében van (például egyetlen R/W vezeték, soronként és oszloponként két-két X és Y vezeték stb.).

A 3. ábrán látható megoldás nagy kapacitású tároknál nem alkalmazható. Ha csak egy 4 k kapacitású tárat képzelünk el, már ebben az esetben is 64 db X és 64 db Y vezeték szükséges. Az ilyen sok kivezetésű áramkört a felhasználók — enyhén szólva — furcsának találnák. A megoldás egyszerű: mivel minden cellához egyetlen meghatározott X-Y vezetékkombináció tartozik, a cellák kiválasztásához alkalmazzuk a (bináris) címet, ebből az IC-tokban elhelyezett ún. címdekódoló áramkörök jelöljék ki a cella kiválasztásához szükséges X és Y vezetéket. Ilyen megoldással, mivel a 4 k, azaz a 4096 rekesz 12 bittel címezhető, mindössze 12 címzővezetékre van szükség.

Tovább csökkenthető — és ez nagyobb kapacitású tároknál célszerű is — a címzésre használt kivezetések száma, ha a tár IC-nek a címet két részletben adjuk: egy 16 bites címnél (64 k) például külön a cím alsó, kisebb helyértékű és felső, nagyobb helyértékű bájtját. Ez a megoldás viszont belső, az IC-ben lévő 16 bites címregiszter és egy vagy két jelzőbit alkalmazását igényli a cím alsó vagy felső bájtjának jelzésére. Csökkenti a kivezetések számát és egyszerűbbé teszi az áramkör alkalmazását az is, ha csak egy-egy információbemenetet és -kimenetet alkalmazunk, függetlenül attól, hogy a mátrixban hány iró-olvasó vezeték van.

A felsorolt megfontolások alapján kialakított és néhány egyéb áramkörrel is ellátott tár rendszervázlatát szemlélteti a *4. ábra*. A vázlat megközelítőleg az általánosan alkalmazott 4164 típusú (64 k×1 bit) tárnak felel meg.

Az egység egytranzisztoros, dinamikus tárolócellákból épül fel. Ennél a rendszernél a kiolvasás töröl, továbbá az információ időnkénti újraírására, frissítésére is szükség van. Ilyen okok miatt szerepel a rendszerben a PUFFER nevű áramkör. A cellakijelölésre a 256×256-os mátrixban a 256 sorvezeték (más kifejezéssel: szóvezeték) és a 256 egyesített iró-olvasó vezeték szolgál. Ez utóbbiak egyben az „oszlop"-kijelölésre is használatosak. Az ábrán nem szerepelnek, de az egységben benne vannak a MOS—TTL és a TTL—MOS szintkonverterek, amelyek a TTL áramkörökhöz való közvetlen csatlakoztatást teszik lehetővé.

Rekeszenként több bitet tároló egységeket több, a 4. ábrán látható rendszer „egymás mögé" helyezésével lehet létrehozni. Az ilyen tárban a szóvezetékek közösek, de az iró-olvasó vezetékek és erősítők száma annyiszorosára növekszik, amennyi az egy rekeszben tárolható bitek száma.

(Folytatjuk.)

NAGY IMRE

A Fővárosi IV. és XV. kerületi Ingatlankezelő Vállalat

felvesz

VI—160 PC IBM kompatibilis mikroszámítógépeihez

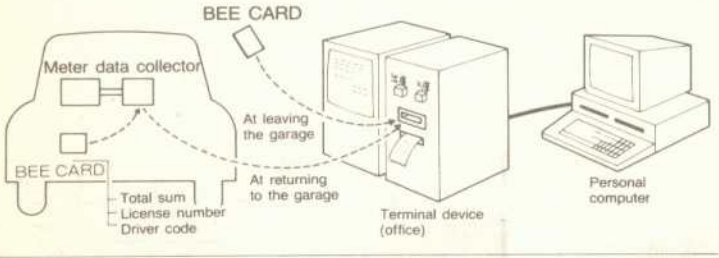
programozókat.

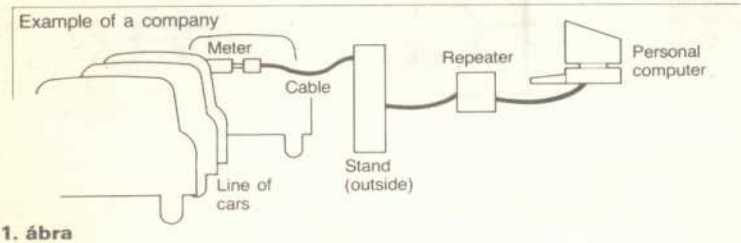
Jelentkezni lehet: részletes önéletrajzzal, személyesen a vállalat Személyzeti és Oktatási Osztályán.

Cím: 1042 Budapest
Munkásotthon u. 66—68. II. 212.

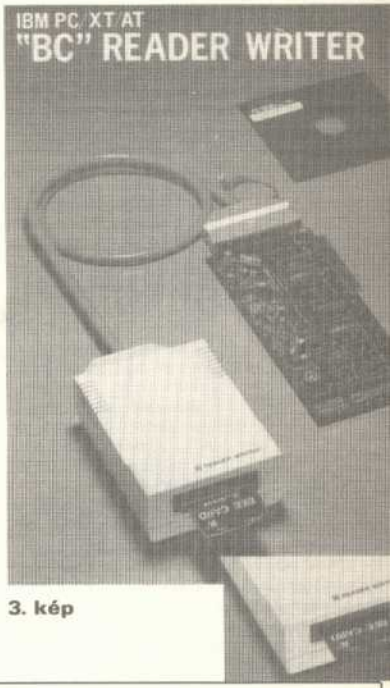
Egy újfajta

Sorozatunkban azokat az új hardver- és szoftvertermékeket ismertetjük, amelyek várhatóan általánosan elterjednek, és meghatározó szerepük lesz a fejlődés irányainak kialakításában.



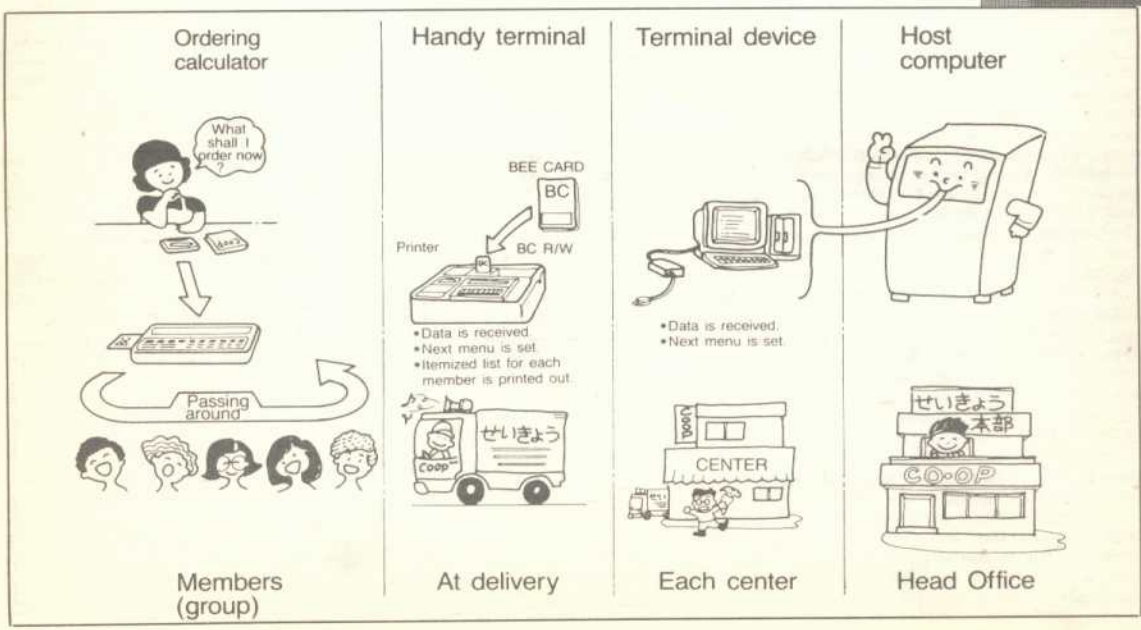


1. ábra



3. kép

2. ábra



nemória

1. kép

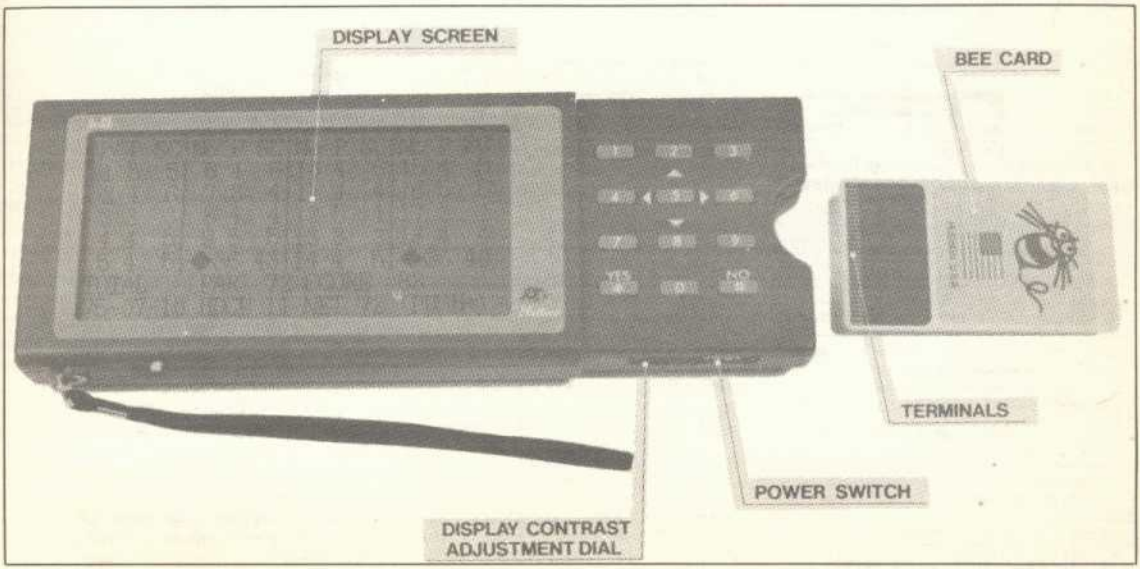
A csekk-kártyák széles körű elterjedése egyre inkább ösztönözte a bűnözőket a velük való visszaélésre, a gyártókat, forgalmazókat pedig a biztonságos csekk-kártyák készítésére. Ebben a harcban születtek meg a memóriás csekk-kártyák. Az *1. táblázatban* az eddig kidolgozott típusokat, a *2. táblázatban* ezek néhány tulajdonságát foglaltuk össze. A *3. táblázatban* a két legelterjedtebb típus, a Bee Card és a CPU-s kártya összehasonlítása látható.

Bee Card

A Mitsubishi konszern Plastics Industries Ltd. vállalata a Hudson Soft Co. céggel együtt fejlesztette ki, vezette be ezt a kártyát. Az *1. kép* a kártya négy típusát (a ROM, az OTP ROM: one-time writable = egyszer írható, az EEP ROM: elektromosan írható és törölhető, valamint az S-RAM kártyát), a *4. táblázat* ezek jellemzőit mutatja. A *2. képen* egy IBM PC változat látható az író-olvasó egységgel.

Az eredeti cél megbízható csekk-kártyák kidolgozása volt, de hamarosan — ahogy a technika történetében már

2. kép



Megnevezés	Szerkezet	Fajta, alkalmazás
Mágneses	mágnescsík	pénzhelyettesítő eszköz (csekk-kártya, telefonhoz is)
IC kártya	CPU-s kártya	ISO típus
	ROM kártya	Bee Card
	RAM kártya	elemes, akkumulátoros
Lézer	NaAg-fóliás	írás után olvasható
	nem NaAg-fóliás, aluminiumtükrözéssel	csak olvasható
Optikai	fotokróm	fejlesztés alatt
	fotokémiai	
	lyukégetés	fejlesztés alatt

1. táblázat. Kártyatípusok

Jellemzők	IC	Lézer	Mágneses
Tárméret	8—256 k	1—4 M	72—144 bájt
Sebesség	gyors, néhány száz ns	lassú, néhány száz ms	nagyon lassú
Ár—teljesítmény	drága, 800—10 000 jen	olcsó, néhány száz jen	nagyon olcsó, 100 jen
Előny	nagyon megbízható	olcsó	nagyon olcsó
Hátrány	érzékeny a feltöltődésre	drága perifériákat igényel, nem írható újra	érzékeny a mágneses mezőkre, nagyon kis tárkapacitású
Alkalmazás	pénzügyi, orvosi, ipari	orvosi, kiadói	csekk és telefon

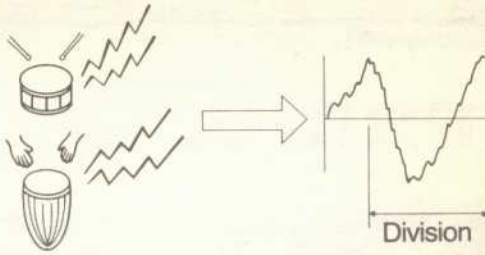
2. táblázat. Kártyák összehasonlítása

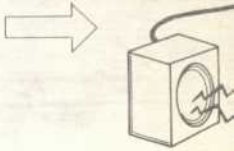
Leírás	hordozható tár	intelligens hordozható tár
Fejlesztési cél	gazdaságos, nagy tárú eszköz létrehozása	a mágnescsíkos kártyákat kiváltó eszköz kifejlesztése
Szerkezet	csak tár	tár és központi egység
Méret	86 x 54 mm	86 x 54 mm
Vastagság	1,8 mm	0,76—0,8 mm
Tüskeszám	32	8
Periféria	író-olvasó, csatlakoztató, Bee R5	különleges író-olvasó
Fajtái	ROM, OTP ROM, EEP ROM, S—RAM	intelligens típusok
Hatásos	gyorsan programozható a párhuzamos csatorna miatt	lassabban programozható és cserélhetők az adatok a soros csatornán át
Szabványosítás	a cikkbeli gyártók javasolták	nemzetközi szabvány van
Ár	gazdaságos, 800—5000 jen	drága, 2000—10 000 jen
Alkalmazás	ipari, orvosi, üzleti, áruforgalmi	főleg pénzügyi

3. táblázat. A Bee Card és a CPU-s kártya összehasonlítása

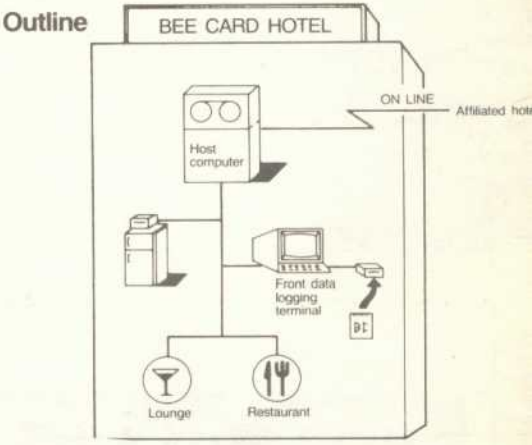
Típus	Tárolásmód	Tárkapacitás	Megjegyzés
Kiolvasható Előnyös, ha azonos adat vagy program van nagy mennyiségben (pl. játékprogram)	ROM	64 k, 128 k, 256 k, 1 M, 2 M, 4 M	
Egyszer írható Bonyolultan írható, ezért sokszori olvasásra jó	EPROM	64 k, 128 k, 256 k, 512 k	2 IC
Újraírható Szabadon programozható, egyszerűen, de lassan	EEP ROM	64 k, 128 k	bájt és teljes lapkiolvasás, 2 IC
Újraírható Gyorsan írható, beépített elemmel	S—RAM	64 k, 128 k, 256 k, 1 M	2 IC

4. táblázat. Bee Card-fajták





3. ábra



4. ábra

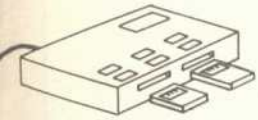
annyiszor — újabb és újabb alkalmazási területeket fedeztek fel. Ezek közül sorolunk fel néhány érdekesebbet:

— Házi vérnyomásmérő. Regisztrálja a mérés időpontját, a magas és alacsony vérnyomás értékét és a pulzust. Mindezt grafikonhoz hasonlóan kirajzolja az orvos számítógépének monitorán.

— Taxik forgalmi ellenőrzője. Regisztrálja a taxi napi forgalmi és műszaki adatait, a taxi azonosítójával együtt. Az adatokat az *1. ábrán* bemutatott rendszerrel lehet a garázsban és a taxivállalatnál feldolgozni.

— Rendelésgyűjtés adatterminálon keresztül. Raktárautomatizálásra,

de Modulation) conversion



BEE CARD

elektronikus számlaellenőrzésre, postai csomagrendelésre használható.

— Egyszerűen cserélhető karakterkészlet mátrixnyomtatókhoz. A karakterkészlet ugyanúgy cserélhető, mint a tárcsás nyomtatóknál.

— Házhoz szállító rendszer. A *2. ábrán* látható rendszerben a rendelőkalkulátorral lehet feladni a rendeléseket, a kártyákat beviszik a raktárakba, ott összeállítják a szállítmányt, majd a központ a raktárak előfeldolgozott adatait átviszi a központi számítógépbe, feldolgozásra.

— Hangforrás hangszintézishez. A *3. ábra* egy dobgépet mutat.

— Szállodai klubtagsági igazolvány. A *4. ábrán* látható rendszerrel például az igénybe vett szolgáltatásokat regisztrálják. Az adatokat a számlakészítésnél és a szálloda gazdasági számításainál használják fel.

CPU-egység

Ugyanezek a cégek készítik a Bee R5 típusú egységet is, amely egy HD6303 (az MC6803, az Építsünk számítógépet! sorozat gépe processzorának kissé módosított változata) típusú mikroprocesszort, 32 k ROM-ot, 16 k RAM-ot, 8 k video RAM-ot tartalmaz. Az egységben 32 × 8 karakteres kijelző (grafikus üzemben 256 × 128 pont), soros csatorna, billentyűzet, energiaforrás és Bee Card csatoló is van *(3. kép)*. Ezt az eszközt Bee Card-adatok energiahálózat nélküli gyűjtésére használják.

SIMONYI

SEMMI SEM LEHETETLEN!

COPYRIGHT MEGRENDELÉSRE

A „Venni vagy elvenni?" mottójú kerekasztal-beszélgetés alapgondolata emésztett, mialatt a Használtcikk Kisszövetkezet SEMMI SEM LEHETETLEN fantázianevű üzletében a Commodore, ZX-Spectrum, Enterprise gépek másolt programkazettáit rakták elém.

De mi is bántott? Egyszerűen szégyenteljesnek tartottam, amikor az NSZK-beli Microsoft cég igazgatója a Magyarországon elterjedt másolásokról kért beszélgetést egy neves magyar cég számítástechnikai konferenciája után. Enyhén szólva nem dicsérte a Magyarországon dúló programmásolást, amit, sajnos, hazánk számítástechnikai üzletpolitikájára nézve is minősítőnek ítélt.

A magyar felszólalók hivatkoztak arra, hogy nálunk is van a szerzői jog védelmét szolgáló jogszabály, és egységesen elítélték a tagadhatatlan kalózkodásokat. Be kell látnunk azonban, hogy a különböző összejöveteleken részt vevő számítástechnikai szakemberek véleménye és állásfoglalása kevés! Nem gondolhattak rá, hogy a jó üzlet, a könnyű haszonszerzés vágya milyen lehetőségeket keres és teremt magának.

Magazinunkban eddig is megragadtunk minden lehetőséget, hogy a szerzők szellemi termékének védelmében szóljunk. Igaz, bizonyos esetekben mi is tehetetlenek vagyunk: a csereberélőket nem ellenőrizhetjük személyenként.

Amit most tapasztaltunk, az minden képzeletet felülmúlt! A jogtalanul programokat másolók hivatalos(?) kereskedelmi forgalomban értékesíthetik portékáikat. A kalózpéldányokat gyári utánzatú kazettaborítóval látják el, tehát nem alkalmi akcióról van szó. Mindezt egy kicsit naiv, de jó üzletember teszi lehetővé.

Hargitai László, a Használtcikk Kisszövetkezet egyik üzletének vezetője első rákérdezésünkre nem is titkolta, hogy másolt játékprogram-kazettákat értékesít. Igaz, a kazettákat alaposan szemügyre véve, ezt nehezen is lehetne kimagyarázni. Kezdeti büszkesége, hogy ő a keresletet és a kínálatot kapcsolja össze és garanciát is ad a kazettákhoz, hamarosan védekezésbe torkollik: „Én csak kazettákat vásárolok! Az, hogy azokon számítógépes program is van, az engem nem érdekel!" — adja elő az így legalizálni vélt kereskedelmi „filozófiáját".

Ennek ellentmond, hogy az üzletvezető a vásárlóknak „katalógust" mutat, amelyből egyhetes szállítási határidővel szinte minden program megrendelhető. Mi ez a minden? Commodore gépre, ZX-Spectrumra minden létező játék. Enterprise gépre körülbelül háromszázötven-féle. Mindez programonként átlag 25 forintba kerül. Honnan ez az olcsóság? A „lopott" holmi általában kevesebbe kerül!

Nézzük csak, miből tevődik össze a gyári program költsége? Szerzői jogdíjból, nyersanyag- és vállalati általános költségből, használati útmutató készítéséből és különféle adókból. És mi a költsége a hivatásszerű másolónak? Nem sok! Szerzői jogdíjat nem fizet, és nem valószínű, hogy adózik. Tagadhatatlan azonban, hogy ügyes. Ugyanis szinte korlátlanul jut hozzá üres, egyébként hiánycikknek számító programkazettához, még olyan minőségben is, amit belföldön nem forgalmaznak. De magyar kazettáról van szó! A kazetta borítója is igen jó minőségű másolást igényel. Vajon mik a beszerzési forrásai? Nem valószínű, hogy saját eszközeivel másol.

Azon már nem is csodálkozhatunk, hogy az Enterprise kazettára védett emblémát másolnak vagy másoltatnak. Ez már igencsak kicsinyesség lenne tőlünk.

Nem utolsó szempont az sem, hogyan kerülhetnek ezek a kazetták hivatalos kereskedelmi forgalomba? Úgy, hogy valaki szándékosan behunyja a szemét. Amiatt persze, hogy túladnak a géptulajdonosok a megunt játékokon, semmi kifogás nem merülhet fel. Az ilyesmi valóban használt cikk. Annyira ellenben nem lehetünk naivak, hogy elhiggyük, egyeseknek korlátlan mennyiségben van ilyen „használt" játékprogramjuk.

Utolsó kérdésünk nem az adóra és a különböző jogszabályokra vonatkozik, hanem így szól: kinek van ebből haszna? A másolónak és a kereskedőnek biztos. És mi történik a géptulajdonosokkal? Rövid távon elképesztően olcsón hozzájuthatnak játékprogramokhoz. Még olyanhoz is, amit érthető „jóérzésből" másutt nem árulnak. És mit veszítenek? Az igazi programkészítők és kereskedők nem foglalkoznak majd programokkal. Nem lesz szoftverfejlesztés és -kereskedelem. Egy régi közmondás szerint nem mindig az a hasznos, ami olcsó!

Az már szinte természetes, hogy az „ügyes" másolókkal nem tudtunk beszélni, mivel nevük és címük — úgymond — üzleti titok.

PINKE GYÖRGY

ADOM A MAGYARÁZATOT!

NEXT without FOR?

Azonos kérdésünk egy programnál bekövetkező két, látszólag indokolatlan eseményre kért magyarázatot: miért ugrik a program egy olyan sorból, ahol nincs is ugróutasitás, és miért adja a cimbeli hibajelzést, ha nincs is hiba?

Az első kérdésre a válasz: a program nem ugrik, hanem csak a TRACE mód kiirása viselkedik „furcsán". Ugyanezt a BASIC-változatot más gépeken használva is tapasztaltam, hogy a FOR . . . NEXT utasitáspárnál ismétlődésnél a FOR utasitást tartalmazó sor sorszámát gyakran „elfelejti". Itt azonban nem irta ki a 6Ø0-as sor után a 63Ø, 67Ø, 6ØØ, 63Ø, 67Ø, 6ØØ, 61Ø, 64Ø, 66Ø, 67Ø, 6ØØ, 61Ø, 62Ø, 63Ø, 67Ø, 69Ø, 67Ø, 68Ø, 14ØØ-as sorokat (ugyanis D=2 volt). Ezzel a jelenséggel eddig még nem találkoztam, de ennél a gépnél ismétléskor ugyanez történt. Más Microsoft BASIC-nél ezt nem tapasztaltam.

A második „hibát" a *listában* található módositás megszünteti. A listából látható, hogy az 14ØØ-as sort a 685-ös sorrá irtuk át (ugyanezt tettük az 138Ø-as sorral, a 655-öst készitettük belőle). Ebből látható, hogy ez a BASIC-változat nem engedi meg a FOR . . . NEXT cikluson belül a GOTO utasitást, még akkor sem, ha az nem jelent kiugrást a ciklusból. Ugyanezt vettem észre más Microsoft BASIC-változatoknál is. A Commodore PC10 tipushoz készitett német nyelvű Microsoft GWBASIC felhasználói leirás (Commodore-kiadvány) az ilyen helyzetet nem tiltja! A Dragon BASIC — amely nagyon közel áll az IBM gépekhez irt Microsoft BASIC-hez — felhasználói kézikönyve szerint GOTO utasitással ki szabad ugrani a ciklusból, de „nem lehet beleugrani a ciklus belsejébe" (idézet az eredeti angol szövegből). Itt mi nem „beleugrunk" ugyan, hanem „benne" ugrunk, de a forditó — úgy látszik — azt nem vizsgálja, hogy honnan ugrunk, csak azt, hogy hová.

SIMONYI

```
6ØØ S=Ø:FOR J=1 TO D:FOR I=1 TO D:E=Ø:IF J=1 THEN 63Ø
61Ø IF IÖJ THEN 64Ø
62Ø FOR K=1 TO J-1:E=E+BúI,KÜ*BúK,JÜ:NEXT K
63Ø BúI,JÜ=AúI,JÜ-E:GOTO 67Ø
64Ø IF I=1 THEN 66Ø
65Ø FOR K=1 TO I-1:E=E+BúI,KU*BúK,JÜ
655 NEXT K:IF BúI,IUÖU2 THEN BúI,IÜ=U2
66Ø BúI,JÜ=úAúI,JÜ-EU/BúI,IÜ
67Ø NEXT I:NEXT J:FOR I=1 TO D:E=Ø:IF I=1 THEN 69Ø
68Ø FOR K=1 TO I-1:E=E+BúI,KU*RúKÜ
685 NEXT K:IF BúI,IUÖU2 THEN BúI,IÜ=U2
69Ø RúIÜ=úLúIÜ-EÜ/BúI,IÜ:NEXT I:FOR I=D TO 1 STEP -1:E=Ø:IF I=D THEN 71Ø
7ØØ FOR K=I+1 TO D:E=E+BúI,KÜ*CúKÜ:NEXT K
```

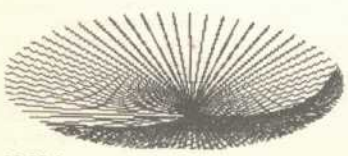
EGY- ÉS KÉTSOROSOK

```
0 PI=3.141593:DEFFNX(A)=(SIN(A)+1)*128:D
EFFNY(A)=(COS(A)+1)*96:A=0:B=0:INPUT"NOV
EKMENY TENYEZOK/KORULFORDULASI TENYEZO (
A,B,T)";AA,BB,T:PMODE4:PCLS:SCREEN1,1:FO
RX=0TOPI*2STEPPI/AA/T:A=A+PI/AA:B=B+PI/B
B:LINE(FNX(A),FNY(A))-(FNX(B),FNY(B)),PS
ET:NEXT
1 IFINKEY$=""THEN1ELSE0
```

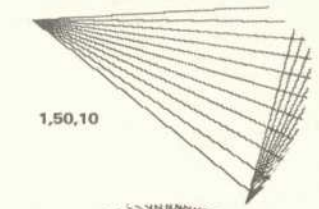
1. lista

```
0 PI=3.141593:DEFFNX(A)=(SIN(A)+1)*128:D
EFFNY(A)=(COS(A)+1)*96:A=0:B=0:FORAA=1TO
100:FORBB=1TO100:FORT=1TO100:PMODE4:PCLS
:SCREEN1,1:FORX=0TOPI*2STEPPI/AA/T:A=A+P
I/AA:B=B+PI/BB:LINE(FNX(A),FNY(A))-(FNX(
B),FNY(B)),PSET:NEXT
1 FORZ=0TO1E3:NEXTZ:NEXTT:NEXTBB:NEXTAA:
STOP
```

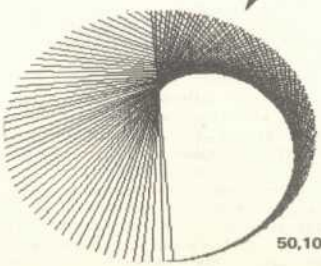
2. lista



32,80,1



1,50,10



50,100,1

A RAINBOW nevű amerikai szaklap minden számában közli az olvasók által bekuldött legérdekesebb egy- és kétsoros programokat. Ezek a programozók valódi ügyességpróbái, hiszen legfeljebb 255 karakternyi helyet kell a lehető legjobban kihasználniuk (a lap a Tandy cég CoCo tipusú számitógépeivel foglalkozik, és az ezekre irt Microsoft BASIC-változat ennyi karaktert fogad el egy sorban). Az már szinte ráadás, hogy ezek a programok gyakran még jól használhatók is.

Az eredeti program magyar üzenettel az *1. listán* látható. Ezt módositottam a *2. lista* szerintire. Igy már a program automatikusan rajzolja az ábrákat. Ezek közül mutatunk be néhány érdekesebbet. A rajzok mellé irt számok jelentése: az első kettő az egyenesek hosszának változását szabályozza, a harmadik a körülfordulások számát határozza meg.

Látható, hogy egészen eltérő tipusú ábrákat is rajzoltathatunk a programmal.

Azoknak, akik a programot saját gépükön kivánják megvalósitani, közöljük az átiráshoz szükséges tudnivalókat. A LINE(X1—Y1)—(X2,Y2), PSET utasitás jelentése: húzz egyenest az (X1,Y1) koordinátájú ponttól az (X2,Y2) koordinátájú pontig. A PMODE4 utasitás a legnagyobb felbontású grafika bekapcsolója. A PCLS letörli a bekapcsolt grafikának megfelelő tárterületet. A SCREEN1,1 a képernyőre viszi a grafikus képet. A LINE—(X2,Y2),PSET utasitás annyiban különbözik az imént ismertetett egyenest rajzolótól, hogy a korábbi egyenesek utolsónak felrajzolt végpontját veszi kiindulási pontnak.

SIMONYI ZSUZSA

µINFORM

Rovatunkban az Apple, Atari, Commodore és Sinclair mikrók tulajdonosait feltehetően érdeklő, angol és német nyelvű cikkekről informáljuk olvasóinkat egy tartalomleíró szólánc segítségével.

A forráshely karaktersorozatát nyíl vezeti be, ezt a / jelig a folyóirat kódja követi (lásd táblázat). A két / jel között a megjelenési adatokat (év, hó), illetve a cikk kezdő oldalszámát szerepeltetjük. A második / jel után pedig - az esetleges másolatkérést megkönnyítendő - a cikk teljes oldalterjedelmét közöljük.

A folyóiratok megtekinthetők a SZÁMALK (Bp. XI. Szakasits Á. u. 68.), illetve - a x-gal jelzettek - az OMIKK (Bp. VIII. Múzeum u. 17.) szakkönyvtárában. (A másolás díja oldalanként 8 Ft.)

A Magazin helyhiány miatt mindössze egyetlen címszó, a programlista közzétételére vállalkozhatott a folyamatosan bővített adatbázisból. A kedvező visszhang alapján az OMIKK háromhavi bontásban kiadja a tartalomleíró szólánc permutálásával és alfabetikus rendezésével szerkesztett teljes anyagot. Az "APACS Mikroindex" első füzete már megjelent, ára 54,- Ft.

Több példány vásárlása/rendelése esetén 10 darabonként 2 tiszteletpéldányt térítésmentesen ajándékoz a terjesztőknek az OMIKK vevőszolgálata (Bp. Pf.: 12. 1428).

A folyóirat neve	Kódja
x 64'er Magazin	64er
Antics	anti
x Chip Magazin	chip
x Compute!	cute
x Dr. Dobb's Journal	dobb
Elektor Electronics	etor
Happy Computer	happ
x mc - Zeitschrift	mc
Run (USA)	run
Run (NSZK)	run2
x Your Computer	your
x ZX Computing Monthly	ZXCM

PROGRAMLISTA
amiga játék (<climber 5>
->cute/87.08-48/3
PROGRAMLISTA
animáció atari xl/xe 3. rész: karakter kép mozgatás ml kódok átalakítása at ascii fuzervaltozokka
->anti/87.08-17/10
PROGRAMLISTA
animáció atari xl/xe játékprogram keszites (<polymove> ->anti/87.08-24/3
PROGRAMLISTA
apple ii grafika hires file-ok tomoritese ->cute/87.08-87/2
PROGRAMLISTA
apple ii játék (<climber 5>
->cute/87.08-51/3
PROGRAMLISTA
atari st (<page flipping> gfa-basic es c nyelvu valtozat
->cute/87.08-72/2
PROGRAMLISTA
atari st grafika trigonometriai fuggvenyek hasznalata demo
->cute/87.08-83/5
PROGRAMLISTA
atari st játék (<tron>
->happ/87.08-61/2
PROGRAMLISTA
atari xl/xe 'autorun.sys' bovites/modositas kepernyo parameterek
->cute/87.08-81/2
PROGRAMLISTA
atari xl/xe (<basic-reset-starter> (<sector copy> ->happ/87.08-60/1
PROGRAMLISTA
atari xl/xe játék (<climber 5>
->cute/87.08-44/4
PROGRAMLISTA
atari xl/xe játék (<diamond dave>
->anti/87.08-25/4
PROGRAMLISTA
atari xl/xe mesterseges intelligencia (<haiku poet> hang/szoveg valtozatok 'rnd' fuggvenyekkel
->anti/87.08-9/6
PROGRAMLISTA
atari xl/xe oktatas(fizika) celsius-fahrenheit-kelvin konverzio
->anti/87.08-36/3
PROGRAMLISTA
c128 c16 c64 programiras 'rem'-kommentarok inverz megjelenitese
->run2/87.08-127/2
PROGRAMLISTA
c16 játék (<airfox>
->run2/87.08-91/4
PROGRAMLISTA
c64 80 karakter/sor editor tv-monitorhoz nem ajanlott! (<64 eighty>
->cute/87.08-76/5
PROGRAMLISTA
c64 filekezeles atstrukturalas kiegeszites a 'datafile 3.6'-hoz (85.11)
->run/87.08-46/6
PROGRAMLISTA
c64 grafika matematika szines mandelbrot kepek eloallitasa
->run2/87.08-120/7
PROGRAMLISTA
c64 játék (<climber 5>
->cute/87.08-47/2
PROGRAMLISTA
c64 játék (hirn 64) 'mastermind' valtozat ->64er/87.08-45/3
PROGRAMLISTA
c64 játék (<pegboard>
->run/87.08-68/3
PROGRAMLISTA
c64 játék (<quadranoid> 'arkanoid'-tipusu jatek ->happ/87.08-47/5
PROGRAMLISTA
c64 karakterkeszlet billentyuk ascii es ernyokodjanak megjelenitese basicbol hivhato 451 byte-os segedlet
->run/87.08-66/2
PROGRAMLISTA
c64 lemezkezeles 'readerror' kereses /kiiktatas olcso lemezeken (<turbo-disk-tester> ->run2/87.08-88/3
PROGRAMLISTA
c64 lemezkezeles (<disk-demon> hibajavitas-visszamentes (0-42) 'killertrack' felismeres ->64er/87.08-37/8
PROGRAMLISTA
c64 lemeztarkezeles katalogusmodositas alfabetikus rendezes (<manager>
->run2/87.08-101/5
PROGRAMLISTA
c64 programiras 8 kepernyo egyideju tarolasa a gep memoriajaban kezelés ctrl+'n' utasitaskeszlettel
->run2/87.08-97/4
PROGRAMLISTA
sprite atari xl/xe játékprogram keszites max. 256 byte-os 'shape' mezok vezerlese basicbol ->cute/87.08-99/3
PROGRAMLISTA
sprite c128 editor-funkciok bovitese
->run2/87.08-94/3
PROGRAMLISTA
szovegfeldolgozas adatatvitel adatbank c64 (david 64) akusztikus kapcsolot es kazettaegyseget is igenylo szoftver ->run2/87.08-106/14
PROGRAMLISTA
szovegfeldolgozas c64 karaktersorozat keresese (<locate string>
->run/87.08-89/2
PROGRAMLISTA
tablazatkeszites atari xl/xe nyomtatas kepforditas szeles formatumokhoz
->anti/87.08-14/7
PROGRAMLISTA
tarszervezes adatbazis c64 60k ram biztositasa 2k-s rezidens programmal
->cute/87.08-93/4
PROGRAMLISTA
tarszervezes c64 gepi kod programiras kezdocim athelyezes (ml relocator>
->cute/87.08-97/3
PROGRAMLISTA
video amiga jatekprogram keszites 3d -hatter-kepek eloallitasa (<fractal-maker> ->run2/87.08-70/12

Minden kedden 17-től 20 óráig
HCC ENTERPRISE klub
a VSZM
Közösségi Házban
(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor
Telefon: 810-950/473

A TUDOMÁNYSZERVEZÉSI
ÉS INFORMATIKAI INTÉZET

előzetes megbeszélés szerint díjmentes programbemutatót tart (vidéken is) az általa forgalmazott oktatóprogramokból.
Horváth Zsuzsa 665-011/2883 mellék
vagy 813-197
Budapest, Pf: 454, 1372

A Schülerdudenről

Nehéz dolog ma olyan informatikai-számítástechnikai lexikont kiadni, amely kiállja az idő — akár csak két-három év — próbáját. A fejlődés ezen a területen olyan gyors, hogy egyes példák vagy akár egész szócikkek is elavulhatnak, mire a mű az olvasóhoz ér. Mégis szükség van olyan könyvekre, amelyekben az ember gyorsan utánanézhet a legalapvetőbb információknak. A Schülerduden tiszteletre méltó vállalkozás még akkor is, ha a rohanó idővel küzdve jó néhány kompromisszumra kényszerültek benne, és a szakemberek esetleg több helyen is vitathatják megfogalmazásait.

A könyv elsősorban az informatikát, számítástechnikát tanuló középiskolásoknak és őket a tanulásban segíteni kívánó szüleiknek, valamint a tanulmányaik kezdetén járó egyetemi hallgatóknak készült. A címszavak kiválasztásánál a szakma pillanatnyi állása és a fejlődés 1986-ban belátható irányain kívül figyelembe vették a német nyelvterület tanterveit és tankönyveit.

Az, hogy a lexikon elsősorban német anyanyelvű **középiskolások** igényei szerint készült, ne tévesszen meg senkit: a hazai **laikus** felnőtt olvasók többsége a szócikkek zömét amúgy is csak számítástechnikai előismeretei birtokában értheti meg kifogástalanul és teljesen.

A könyv praktikus és szép kivitelű, a szövegbe jól illeszkednek a főleg Pascalban írt programozási példák. S természetesen ott jó, ahol a számítástudomány már kiforrott(abb), konszolidálódottabb eredményeire, alapjaira támaszkodik, vagy éppen azokat mutatja be.

A címszavak (van belőlük!) kiválasztásával egyetérthetünk, a szócikkek hosszának arányaival nem mindig. Érthetetlen például, miért kapott az **Editor** címszó 13 oldalt, a **Digitalrechner** — digitális számítógép — viszont mindössze 24 sort. A **Computer** szócikk is csupán öt és fél oldal hosszúságú.

A lexikonon — érthetően — érezhető egyfajta *„definíciós kényszer"*, amelynek határt szab például a könyv mérete. Itt-ott talán engedményt lehetett volna tenni e tekintetben, és nyugodtan elhagyható lett volna a „definíció" olyan szócikkek elejéről, ahol a *címszó* önmagáért beszél. Például a **Datenübertragung** címszónál az, hogy az **adatátvitel:** az „adatok átvitele két vagy több, térben egymástól tetszőleges távolságban lévő eszköz — adatállomás — között".

A német nyelvű kiadás (1986) óta eltelt két év alatt is nagyot lépett előre a technika. Megnőttek például a tipikus tárméretek mind a félvezető chipeknél, mind a mágneses és egyéb adathordozóknál; más mikroprocesszor-típusok vannak terjedőben stb.

Tudomásunk szerint tervezik a lexikon **magyar nyelvű kiadását,** aminél több dolgot is jó lenne figyelembe venni. Például:

— a magyar diákok számítástechnikai-elektronikai előismeretei eltérhetnek attól, amit a Duden a német, svájci, osztrák(?) diákoknál az ottani tantervek és tankönyvek alapján feltételez;

— a Duden — érthetően — sok helyen az NSZK-beli környezetet veszi alapul, többek között a **Bundespost** előírásait. A magyar kiadásnál természetesen a Magyar Postára, illetve, ha vannak, a magyar szabványokra (is?) kellene hivatkozni;

— a **„napi valóságot"** érintő szócikkeknél figyelembe kellene venni az elmúlt két évben bekövetkezett változásokat;

— azokon a helyeken, ahol a „definíció" nem mond többet a címszónál, el lehetne tekinteni tőle;

— az elméleti alapokat érintő szócikkeket több helyen — megadva a tiszteletet az eredeti szövegnek — feszesebbre lehetne fogalmazni;

— célszerű lenne kiegészíteni a lexikont néhány jellegzetes, magyar vonatkozású szócikkel is. (Például K. Zuse mellett meg lehetne említeni Neumann Jánost is stb.);

— néhány helyen, például az említett **Editor** szócikknél javítani lehetne a terjedelmi arányokon.

A **„besserwisserkedést"** elkerülve, a fordítással itt-ott mindenképpen járó **szövegromlást** minimálisra csökkentve — ügyelve a szép, érthető, **magyaros** fogalmazásra —, a változtatásokat az eredeti mű kiadójával természetesen egyeztetve, érdemes a könyvet magyarul is *minél hamarabb* kiadni. Még akkor is, ha mások is észlelik majd azt, amit e rovat írója saját magán észrevett: örült a könyvnek, szívesen vette a kezébe, szívesen forgatta, mint olvasmányt, de amikor *„nagyító alá vette"*, sokszor bosszankodott is. A maga részéről kompromisszumot javasol: vegyük át és fogadjuk el a Dudent a fenti szempontok szerinti *kis* változtatásokkal annak, ami a rendeltetése: bárhol felüthető *szakmai* olvasókönyvnek, és ne tekintsük *orákulumnak*, kutatási alapműnek, megfellebbezhetetlen hivatkozási alapnak. Örüljünk, hogy legalább ez forgatható lesz magyarul is.

Egyezzünk meg továbbá abban: ne tekintsük ezt a lexikont *céltáblának* se. Ha ez a könyv nem lesz elég jó nekünk, akkor csináljunk magunknak, diákjainknak jobbat — ha tudunk.

Ez a rovat nem lenne hű önmagához, ha a Schülerdudenről szólva ennyivel beérné — sőt fennállásának immár két és fél éve alatt nem is adott még ilyesfajta könyvismertetést. Hagyományainkhoz híven tehát villantsunk fel most is kedvcsinálónak egy parányi részletet a könyvből.

A **Künstliche Intelligenz** (magyarul: mesterséges intelligencia) címszónál találtunk rá az alábbi gondolatra: Azok a problémák, melyek megoldásához intelligenciára van szükség, rendkívül komplexek, és a megoldás nem adható meg áttekinthető algoritmusok formájában. Tipikusan ilyenek a stratégiát igénylő játékok — különösen a *dáma* és a *sakk*. Ha egy sakkjátékos öt teljes lépés (tíz ún. féllépés) figyelembevételével kívánná a következő lépést kiválasztani az összes lehetőség szisztematikus számbavételével, akkor $25^{10} = 9{,}5 \times 10^{13}$ állást kellene kiértékelnie. Ha ezt a munkát egy olyan — különlegesen gyors — géppel végeztetnénk el, amely másodpercenként 250 000 állást lenne képes kiértékelni, akkor ennek a gépnek több mint tizenkét évig kellene egy-egy újabb lépésen „gondolkoznia".

A mesterséges intelligencia egyik tipikus kérdése az, mit lehet a problémával akkor kezdeni, ha a lehetséges választások száma már áttekinthetetlenül nagy. Az *intelligens* ember — például egy sakkjátékos — szemben a *nem* intelligens géppel, *tud* mit kezdeni ezekkel a feladatokkal: azért intelligens.

Hogyan ábrázolható (írható le) a tudás? A tudás egyrészt speciális részletekre vonatkozik, másrészt általánosan érvényes információ is. Ha tudom például valakiről, hogy van színes televíziója, akkor tudom azt is róla, hogy televíziójában van színes képcső is, a készüléken vannak kezelőszervek, de azt már például nem feltétlenül tudom, hogy készüléke milyen márkájú, telepről is működik-e vagy csak hálózatról, mekkora a képcső mérete. A *részletekre* vonatkozó tudásomat ún. *tények*ként, az *általánosítható* információt ún. *szabályok*ként kezelem. (Lásd például a PROLOG nyelvet.)

Hogyan lehet már meglévő tudásból új tudást nyerni? Ez is a mesterséges intelligencia kérdéskörébe tartozik. A Duden példája szerint ha valakiről tudjuk, hogy a teniszszövetség elnöke, akkor ebből a tudásból következtetni tudunk arra is, hogy az illető *tud teniszezni*. Van tehát valami olyan tudás is, amit mondjuk *„meta-tudás"*-nak nevezhetnénk. Ezt az ún. *meta-tudást következtetési szabályokba* (Inferenzregeln) foglaljuk.

Befejezve a kiemelt részlet bemutatását, Nyugatra barangoló turistáinknak és a német számítástechnikai szaknyelvet tanulóknak is ajánljuk: külföldi útjaikon járva ne csak gépeket vegyenek, hanem szakkönyveket is — például éppen ezt a Schülerdudent.

Akik a Mikroszámítógép Magazint rendszeresen olvassák, biztosan észrevették már, hogy e rovat az 1988/8. számtól kezdve „Updike-korszakába" lépett. A *Feladványok*, a *Passiánsz* és *Az ártatlanság bolygója* után most következzék:

J. UPDIKE

A legutóbbi ülés jegyzőkönyve

A Bizottság elnöke bejelentette lemondási szándékát. A titkár felhívta a figyelmet arra, hogy az alapszabályokban nincs szó lemondási eljárásról, van viszont új tisztségviselők évenkénti jelöléséről, amit most annak rendje és módja szerint lefolytatnak.

Az elnök azt válaszolta, hogy az új listán ismét elnöknek jelölik. Elmondta, a Bizottság alapítása óta tölti be ezt a tisztséget, és őszintén az a véleménye, hogy elnökösködése most már inkább akadály, mint segítség. Szerinte amire a Bizottságnak most szüksége lenne, az az új irányítás és a

egy informatikai lexikont (Schülerduden, Die Informatik, Bibliografisches Institut, Mannheim/Wien/Zürich, Dudenverlag, 1986. 538 old.) és egy Updike novellát (Minutes of the Last Meeting). Szokásunkhoz híven élményünket önökkel is megosztjuk.

célok iránti nagyobb érzékenység, amivel ő nem szolgálhat, mert túl öreg és szórakozott már, elvesztette érdeklődését a dolgok iránt. Elérkezett az ideje annak, hogy valaki, aki fiatalabb, vegye át a kormányrudat. Vagy az is elképzelhető, hogy esetleg feloszlassák a Bizottságot.

A titkár felhívta a figyelmet arra, hogy az alapszabályok a feloszlatásról nem intézkednek.

Az elnök felszólalására válaszolva Mrs. Hepple, a Jelölő Albizottság képviseletében elmagyarázta, hogy az Albizottság egyöntetűen úgy érezte, az elnök személye nélkülözhetetlen jelenlegi pozíciójában, mert a szélesebb közösség támogatása lemondásával drasztikusan csökkenne. Két alelnök kinevezésével és megfelelő albizottságok létrehozásával egyébként hatásosan csökkenteni lehetne az elnök terhelését.

Az elnök megkérdezte, hányszor ült össze a Jelölő Albizottság. Mrs. Hepple azt válaszolta, hogy a nyári szabadságok miatt mindössze egyszer, telefonon. Erre kitört a nevetés. Az elnök ugyanilyen humoros szellemben azt mondta, hogy visszavonulására ilyenformán csupán egyetlen lehetőséget lát, ha főbe lövi magát.

Mr. Langbehn, az egyik újsütetű tag azt mondta, mielőtt eldöntené, részt vesz-e ebben a vitában, hálás lenne, ha valaki elmagyarázná neki a Bizottság eredeti célját és szándékait.

Az elnök azt válaszolta, ő maga sohasem értette ezeket, ő maga is nagyon hálás lenne a magyarázatért.

Miss Beam jelentkezett, hogy bár a jelen lévő alapító tagok közül ő a legfiatalabb, közre tudná bocsátani emlékeit, miszerint a Bizottság alapításakor az volt a formális cél, hogy jóváhagyják az igazgató tevékenységi körét. Az igazgató varázslatos személyisége és mély elkötelezettsége nélkül össze sem gyűltek volna, és azonkívül, hogy kinevezték az igazgatót, az ülés figyelmének oroszlánrészét a Bizottság nevének megválasztására koncentrálta. A kiinduló javaslat „Tarboxi Közjavítási Bizottság" volt, amit aztán kibővítettek az „Emberi Erőforrások Megjavításának és Továbbfejlesztésének Bizottsága" elnevezésre. Úgy emlékszik, az igazgatónak az volt az érzése, az „esélyegyenlőség" kifejezést is be kellene venni az elnevezésbe, és talán még valamilyen módon az ifjúságnak is hangsúlyt kellene adni anélkül, hogy ezzel kizárnák a közösség idősebb polgárait. Így aztán javaslatba és fontolóra vétetett a „Bizottság Tarboxi Fiatalok és Öregek Egyenlő Esélyeiért és Fejlesztéséért" elnevezés.

Dr. Costopoulos, az egyik alapító, emlékeztetett arra, hogy az igazgató nem kívánta a Bizottságot olyan színben feltüntetni, mintha az rivalizálni szándékozna olyan, már létező csoportosulásokkal, mint az „ÖTYE" vagy a „TIZI-BIZI", továbbá helytelenített volna a Bizottság elnevezésében bárminemű utalást az elszemtelenedő környezetvédelemmel való azonosulásra. Így egyhangú szavazással a Bizottság nevének kérdését ideiglenesen nyitva hagyták.

Mrs. MacMillan, az egyik új tag, megkérdezte, hol van most az igazgató.

Miss Beam elmagyarázta, hogy az igazgató az alapító ülés után eltűnt. Egy kartonbőröndöt és egy kifizetetlen telefonszámlát hagyva maga után — szólt közbe az elnök. Kitört a nevetés.

A titkár rámutatott arra, hogy az alapszabályok teljesen világosan meghatározzák a Bizottság célját, és kivonatokat olvasott fel olyan bűvmondatokkal, mint „tilos mindenféle politikai jelölt vagy partizánügy nyilvános támogatása", „tilos részvények vagy kötvények tartása pénzügyi haszonszerzés vagy nyereség céljából" és „a Bizottság által részben vagy egészében birtokolt vagy bérelt helyiségekben tilos mindenféle szerencsejáték vagy a jó erkölcsbe ütköző cselekmény".

Mr. Langbehn kérdezte: megnézhetné-e az alapszabályzatot. A titkár készségesen átnyújtotta. Mr. Langbehn némi tanulmányozás után azt állította, hogy az nem más, mint minden irodaellátónál vagy papírüzletben kapható formanyomtatványok halmaza.

Mrs. Hepple azt mondta, nem látja be, ennek mi lenne a jelentősége, az a lényeg, hogy valamennyien itt vannak.

Mrs. MacMillan azt tudakolta, miért ülésezik a Bizottság az igazgató távollétében.

A pénztáros közbeszólt és azt kérdezte az est háziasszonyától, Mrs. Landistól, nem lenne-e itt az ideje a frissítők felszolgálásának.

A nagytiszteletű Mr. Trussel azt kérdezte, megvilágíthatná-e a jó Mrs. MacMillan által felvetett kérdést. Azt mondta, a Bizottság kezdetben abban a reményben ült össze, hogy az igazgató ismét megjelenik. Később pedig, mint annak kivizsgálására alakult testület, hogy hová mehetett az igazgató. Végül azért folytatták az ülésezést, mert — véleménye szerint — megszerették egymást és igényelték egymás társaságát.

Miss Beam úgy gondolta, hogy amit hallottak, az megindító és találó magyarázat.

Mrs. Hepple azt mondta, nem látja be, miért lenne ez itt most lényeges, mivel nemcsak az alapító tagok többsége van jelen, hanem sok új tag is. A taglétszám éppenséggel nőtt, ahelyett, hogy csökkent volna, ahogy az ember akkor feltételezné, ha a Bizottság az igazgató személyétől függene, akiről egyébként már azt is elfelejtette, milyen volt.

Mr. de Muth jelentkezett, hogy ő annak idején mint szociológiatanár lépett be, mert úgy értesült, fontolóra vették egy program indítását, melynek keretében az iskolák erőforrásai javításának témájában előadás- vagy happeningsorozatokat rendeztek volna.

Mr. Tjadel annak idején mint a fák sebészorvosa lépett be, „környezetvédelmi szempontból".

Mrs. MacMillan azt mondta, hogy neki azt mondták, szájon át szedhető fogamzásgátlók városi ivóvízbe keverése iránti érdeklődése hasznára lehet a bizottságnak.

Az elnök rámutatott, hogy az egész marhaság, és ismét felajánlotta lemondását.

A titkár felhívta a figyelmet arra, hogy a fent említett javaslatokat és terveket mind fontolóra vették, és — amennyire ő tudja — ezeket ma is napirenden tartják.

Mr. Langbehn köszönetet mondott a jelenlevőknek, hogy vele tartanak és oly nagyszerűen gondoskodnak táplálásáról. Azt mondta, bár a felsorolt tervek egyike sem hozott idáig semmi kézzelfogható eredményt, nem érzi úgy, hogy — saját magát is beleértve — a Bizottság tagjainak szégyenkezniük kellene amiatt, hogy fáradozásaik eddig hiábavalók voltak. Éppen ellenkezőleg, vitákat kavartak és felkeltették a szélesebb közösség érdeklődését, továbbá azáltal, hogy olyan kiváló és köztiszteletben álló személyiségeket vonzottak magukhoz, mint a jelenlevők, a kudarcnak még az árnyéka sem vetődhet rájuk. (Itt hiányzik néhány mondat, mert egy pohár véletlenül felborult. — A titk.) Azt mondta, amire szükség van, az nem az igazgató által kijelölt horizont bárminemű hosszabb távú szűkítése, hanem csupán az erők pillanatnyi koncentrálása néhány kétségkívül korlátozottabb, ámde a szűkebb közösségen belül megvalósítható rövid távú cél meghatározására.

Mrs. Hepple azt javasolta, rendezzenek táncestélyt vagy kirakodóvásárt abból a célból, hogy pénzalapot gyűjtsenek ilyesfajta célok megvalósítására.

Miss Beam úgy gondolta, utcabál jobb lenne, mint táncestély, ez jobban vonzaná a fiatalságot.

Az elnök azt indítványozta, módosítsák az alapszabályokat úgy, hogy lehetővé tegye a Bizottság feloszlatását. Senki sem támogatta ezt az indítványt.

Mr. Tjadel azt mondta, nem látja be, miért aggódtak annak idején az emberi erőforrásokat illetően, amikor véleménye szerint már ebben a teremben is olyan kitűnő emberi „erőforrások" gyűltek össze. Erre kitört a nevetés és a taps.

A pénztáros jelentkezett azzal, hogy véleménye szerint eddig ez volt a legjobb ülés, és a Bizottság azzal tehetné munkáját még hatékonyabbá, ha gyakrabban ülésezne.

Miss Beam azt mondta, együtt érez az elnökkel, és úgy gondolja, tisztelet ben kellene tartani az elnök óhaját.

A nagytiszteletű Mr. Tjadel azt indítványozta, hogy a beterjesztett formában, azzal a megszorítással fogadják el a Jelölő Albizottság javaslatát a tisztségviselőkre, hogy az elnököt átmeneti időre nevezik ki, és munkájának megsegítésére hozzanak létre egy újabb „Célok és Szándékok Albizottságot", Mr. Langbehn és Miss Beam alelnökletével az élén.

Mrs. Hepple és mások támogatták az indítványt.

A szavazás eredménye egyhangú volt, az elnök tartózkodott a szavazástól.

(Fordította: — KE —)

From the book Problems and other stories by John Updike. Copyright C 1971, 1972, 1975, 1983 by John Updike. „Minutes of the last Meeting" originally appeared in Audience.

A SAKKPROGRAMOZÓKAT SEGÍTŐ STATISZTIKAI ADATOK

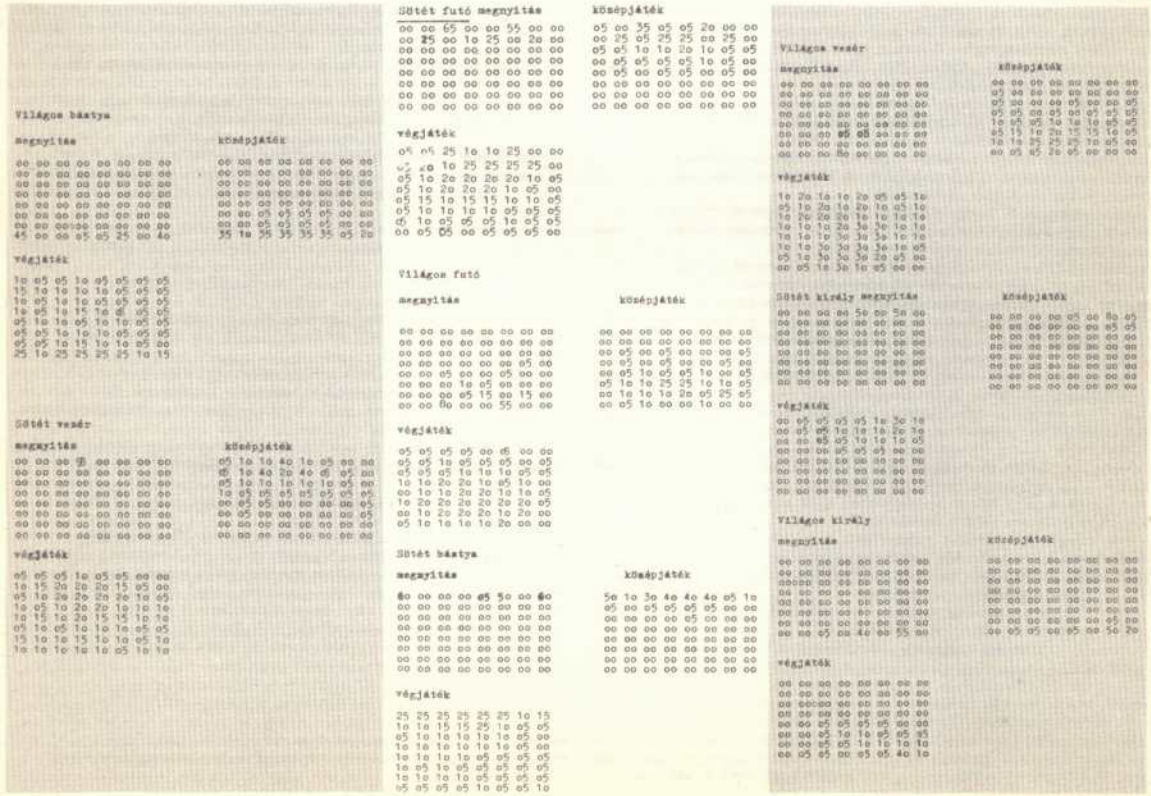
Az előző alkalommal a gyalog és a huszár elhelyezkedésével foglalkoztunk. Most ismét a holland Dap Hartman adatait ismertetem, amelyek megmutatják, hogy a játszma megnyitásában, a középjátékban és a végjátékban az egyes mezőkön milyen valószínűséggel fordul elő futó, bástya, vezér és király. Az adatok kiszámításánál a szerző a már ismert 832 parti pozícióit vette figyelembe, amelyekben 62 965 különböző állás jött létre.

A következő táblázatok megmutatják, hogy az egyes figurák hány százalékos valószínűséggel találhatók meg az egyes mezőkön a játszma három szakaszában, a megnyitásban, a középjátékban és a végjátékban. Az egyszerűség kedvéért az adatokat feldolgozó program a harmincadik félépésig tekintette az állást megnyitásnak, középjáteknak pedig azokat a pozíciókat számította, amelyekben a tisztek száma — a királyokat nem tekintve — hatnál nagyobb. A többi esetben az értékelő algoritmus az állást mint végjátékot dolgozta fel.

A futókra a táblázat a következőképpen alakul:

A táblázatok számadatai jól szemléltetik, hogy a mestererősségű játékosok a játszma egyes szakaszaiban mely figuráikat aktivizálják és azokat melyik mezőn helyezik el. A királyokra vonatkozó táblázat megmutatja, hogy az erős játékosok szinte kivétel nélkül rövidre sáncolnak, a végjátékban pedig nagyon figyelnek arra, hogy az ellenfél királya ne tudjon betörni térfelükre. A vezér a megnyitásban és a középjátékban a tábla szélén helyezkedik el, onnan ellenőrzi a centrummezőket, és csak a végjátékban merészkedik ki a tábla közepére.

KOVÁCS P. ATTILA



Az Olvasó írja

Egyre több a vállalkozás a szakmában. Most két diákvállalkozásról adok hírt. Nagyon remélem, hogy a példát mások is követni fogják.

Rábel Csaba, Kiskunhalas

Úgy hallottam, hogy nemcsak nagy terjedelmű programok kellenek az olvasóknak, és én is szeretnék azok közé tartozni, akik megváltoztatják az eddigi helyzetet. Nem sok program jelenik meg lapjukban, aminek rendszeres olvasója lettem az 1985. évi bajai számítástechnikai tábort követően.

Szeretnék vállalkozni arra, hogy havonta egy programmal segítem a Magazint. Édesapám azt mondta, hogy eddig csak hobbim volt a számítógép, csináljak belőle egy kis hasznot. De lényegében nem ez a fontos, hanem az, hogy lehetőleg sokan fel tudják használni kisebb-nagyobb programjainkat.

A GTV-nkről röviden: GTV C. A. R. S. a teljes név, amely iskolánkról kapta nevét: Garbai Sándor I. SZ. I. Garbai Television Computer and Audiovisual Recording Studio. Mi számítógépekkel (C64, C128, Plus/4, Enterprise) feliratokat és reklámanyagokat készítünk a városi kábeltelevízió részére. Természetesen a tagok egymás között csereberélik programjaikat. Minden tagnak van egy betűjele. Nekem @ GTV/a, másnak más. Ketten vezetjük a társaságot, Mátyási Arnold és én. Hozzánk futnának be a programok s mi küldenénk be a szerkesztőségnek elbírálásra.

Szeretném tudni, hogy beküldhetünk-e programokat? Mennyi az átfutási idő a levél érkezése és a kinyomtatás között?

Köszönöm a vállalásukat, a havi egy programot szeretettel várjuk. Azt tudom ígérni, hogy a jó és szellemes munkákat közölni fogjuk. A beküldés és megjelenés között hónapok telnek el, még akkor is, ha a kapott program zseniális, tudniillik a lap átfutási ideje mintegy 3 hónap.

Sok sikert kívánok a CARS-nak, de mindjárt meg is kérdezem, hogy miért kell egy bajai klubnak angol nevet adni?! Ezt tegyék az angolok Londonban. Ha egy kicsit törik a fejüket, biztosan találnak jó és szellemes magyar elnevezést is. Még egyszer sok sikert!

Sorosy Tamás, Székesfehérvár

Több mint két éve számítógép-tulajdonos vagyok, és érdeklődve böngészem különböző rovataikat. Az 1988/8. szám Az olvasó írja sorai között bukkantam rá Kelemen Mihály levelének töredékére, ahol TVC-re keres olyan másolót, melyben kimentés előtt megváltoztatható a program neve. Nekem van egy ilyen, saját céljaimra fejlesztettem ki itthon. A másoló még teljesen „friss", két hete készült el. Extrái között megtalálható az a funkció is, melyről Kelemen Mihály ír. Éppúgy lehet dolgozni vele kazettára, kazettáról, mint lemezre és/vagy lemezről. A gép programjainak nagy részét a Novotrade forgalmazza. Levelemmel az önök útján szeretnék kapcsolatba kerülni velük, hogy programomat értékesíthessem, esetleg úgy, hogy a magazin is jól járjon a másolóval. Úgy tudom, ilyen kaliberű másoló még nem készült TVC-re. Ezért kérném önöket, hogy a Novotrade megfelelő szekciójába továbbítsák levelem. Kérem, közöljék velem, ha a Novotrade érdeklődik a program iránt, és azt is, ha nem. Fáradozásukat előre is köszönöm: Europe Software Company.

Nem szeretném visszaháritani a Novotrade-del való üzleti kapcsolat létesitését, de a szakmában úgy szokás, ha valaki el akar adni valamit, az ir egy levelet a feltételezett vevőnek, elmondja a feltételeket, és felkinálja az árut. Ha már közvetitőt vesz igénybe, az rendszerint le is csip valamit a vételárból. (A Mikroszámitógép Magazin persze nem ilyen szervezet!) Szóval az a javaslatom, hogy a következő cimre küldjön levelet: Novotrade Rt., Budapest 1389 62. Pf. 139., és irja le mindazt, amit nekünk is megirt. Az üzlet komolyságát növelni fogja, ha a nevét irja alá és nem azt, hogy Europe Software Company, mint azt velünk tette. Thanks!

KOVÁCS GYŐZŐ

ÁTALÁNYDÍJAS JAVÍTÁSI ÉS KARBANTARTÁSI SZERZŐDÉS = ÖRÖK GARANCIA

FOTOELEKTRONIK ALFA GT NOVOTRADE

COMMODORE, ATARI, TVC stb. személyi számítógépek, valamint IBM PC/XT/AT professzionális PC számítógépek és perifériák (floppy, printer) garanciális és fizető JAVÍTÁSA, KARBANTARTÁSA.

SZERVIZEINK:

1053 Budapest V., Magyar u. 12–14. T.: 173-551 Tx: 7621
1083 Budapest VIII., Szigony u. 9. T.: 343-153
1191 Budapest XIX., Gábor A. sétány 3.
3100 Salgótarján, Arany János u. 3. T.: 32-14-007
3525 Miskolc, Fazekas u. 1–3. T.: 46-17-011
4034 Debrecen, Holló László u. 14.
5600 Békéscsaba, Bartók Béla u. 37. T.: 66-27-195
6726 Szeged, Székelysor 13. T.: 62-13-377
7400 Kaposvár, Füredi u. 24. T.: 82-16-307
7624 Pécs, Jurisics M. u. 17. T.: 72-11-812
9700 Szombathely, Szalonok u. 31. T.: 94-14-519

1053 Budapest, Henszlmann I. u. 9. Telefon: 174-144 Telex: 22-7621
VEVŐSZOLGÁLAT — FOTOELEKTRONIK — NOVOTRADE — ALFA G.T.
VÉTEL — ELADÁS 1077 Budapest, Dohány u. 16. Tel.: 428-936

TII
SZELVÉNY
MIKROSZÁMÍTÓGÉP
MAGAZIN
1988.
NOVEMBER

Egy sarokkal olcsóbb!

Lányok, fiúk! Még kedvenc tantárgyaitok is kifognak rajtatok olykorolykor. De ne csüggedjetek, ne búsuljatok. A Tudományszervezési és Informatikai Intézet Videoton TV—Computeren futtatható, kedvezményesen megvásárolható oktatóprogramjaival megkönnyíti az otthoni fejtágítást. A programok jobban „súgnak", mint a legügyesebb padtárs. Tábla hiján a monitoron mutatják érzékletesen a tananyagot, és addig nyúzhatjátok magatokat példákkal, amíg ki nem okosodtatok. Ráadásul a bonyolult számokkal, szerkesztésekkel sem kell papíron bajlódnotok. Ennyi segítséggel a szigorú szülői kikérdezés már igazán csak gyerekjáték!

Aki tehát (szülő is lehet) a lapunkból kivágott sarokszelvényt a TII-ben (Budapest XI., Egry József u. 1—9. BME „E" épület XI. em. 111. Postacím: 1372 Bp. Pf. 454) átadja, vagy megrendelésével együtt elküldi oda, havonta húsz oktatóprogram közül válogathat, olcsóbban juthat hozzájuk. A kedvezmény a szelvényen feltüntetett hónapban érvényes.

Postai megrendeléskor (a megrendelőlapon legyen rajta a név, pontos cím, személyi szám, dátum) programjainkat utánvéttel szállítjuk, amire 6 hónap garanciát vállalunk.

Az e havi kedvezmény

Matematikához a Számelmélet, Egyenlet, Kombinatorika, Függvényábrázolás és transzformáció, Gyökkereső, Galton, Prim programokat, fizikához a Soros RC körben lejátszódó folyamatok modellezése, Mágneses mező, kémiához a Mend, biológiához a Növényhatározás, idegen nyelv tanulásához az Ürcsata, Magyar—orosz szótár, Interaktív magyar—angol és magyar—német szótárprogramokat, földrajz tanulásához a Magyarország, Szovjetunió és USA, s mindezeken kívül a Ki mit tud? és az UNIFEL programot ajánljuk.

Egy program ára 313 forint

Majdnem egy kis sült galamb röpül a szátokba, ha a Gyökkereső programba kóstoltok. Persze azért definiálnotok kell egy egyváltozós függvényt, s a program máris rajzolja annak ábráját, a szintén általatok megadott intervallumban. No és természetesen megkeresi a függvény gyökhelyeit. Figyelem, azért ezt a kényelmet nem adják olyan olcsón! A függvény definiálásához a semminél több tudás kell BASIC-ből. A Galton-deszka kísérletet szimuláló programmal nyomon követhetitek az egyes szinteken az empirikus eloszlás változását, annak konvergenciáját az elméleti eloszláshoz. Ezenkívül hisztogram is készül a választott szint empirikus és elméleti eloszlásáról. Mi más is lehetne a program neve, mint Galton, mellyel a negyedikes gimnazisták fakultatív matematikaoktatása tehető szemléletesebbé.

Nem hagyjuk cserben a középiskolás elsősöket sem. A ti matematikai fejtágításotokhoz hozzá tartoznak a számelméleti alapfogalmak. Erre „hangolódik" a Prim program, melylyel a — nomen est omen — prímszámok keresésének algoritmusát láthatjátok a képernyőn az „Erathosztenész rosta" alapján. Ez három részből áll. Az elsőben megfigyelhető a képernyőn a 257-ig terjedő prímszámok kirostálása az Erathosztenész-féle szitával. A másodikban leolvashatók a prímszámok maximum 6000-ig. Végül az adott intervallumba eső prímszámokat keresi ki a program, s írja a monitorra. Ez utóbbi rész felső határa legfeljebb 32 768 lehet.

Az USA program az Amerikai Egyesült Államok 45 államának fővárosát ismerteti név és hely szerint. A cseppet sem könnyű gyakorlásban hasznos fogódzóul szolgál.

Mágnessel játszani a látványossága miatt nagyon szórakoztató. Ám ennek a jelenségnek fizikai összefüggéseit megérteni már lényegesen bonyolultabb. A Mágneses mező program az elektromos vezető környezetében létrejövő mágneses mező vizsgálatát és számítását készíti el.

A két Interaktív szótárprogram a magyar—angol és a magyar—német nyelvtudást segíti elő. A betáplált angol—magyar — illetve a másik idegen nyelv — szópárok kiírásával könnyíti az új kifejezések emlékezetbe vésését, továbbá játékosan ellenőrzi a diák lexikális tudását.

Végezetül ajánljuk az igen hasznos, érdekes Ki mit tud? programot. Az ötletes szoftver iskolai vetélkedők, vagy éppen sportversenyek döntőjének eredményét vetíti a képernyőre. Ugyanakkor a tömbök méretének változtatásaival tanulmányi eredmények, átlagok jegyzésére is alkalmas.

EGY KIS ELEKTROSZTATIKA

„Oly távol vagy tőlem, és mégis közel" — hallottuk a dalban a telepátia ismérveit. A laikusok többségének azonban még máig sem igazán érthető, miért is vonzzák vagy taszítják egymást az elektromossággal töltött testek. Mindenesetre rajtuk az igen alapos magyarázattal szolgáló kvantum-elektrodinamika alig segíthet, mert egy laikus számára képtelenség úgy leegyszerűsíteni az elméletet, hogy pofonegyszerű legyen a magyarázat. Sokakban — még a képzettekben is — fel sem merül olyan kétség, hogy itt valami sötét pont van, hiszen az iskolában megtanulták: két test távolból is tud hatni egymásra. Pedig ha tudnák, hogy Newton milyen boldogtalan volt amiatt, hogy életében nem tudott magyarázatot adni a testek „vonzóerejére"! Hol az a láthatatlan kéz, amellyel a Föld, a Hold, a Nap, a bolygók, a csillagok egymásba fogódzkodva tudnak járni égi országútjaikon?

Be kell vallani, hogy a gravitációra még ma sincs igazi fizikai magyarázatunk, jóllehet a Magyari-féle potentonelmélettel még kiemelkedő magyar versenyző is szerepelt századunk elején a megoldásért vivott csatában. (Magyari éppen a ma már múzeumi értékű Laki-hegyi adótornyot használta az elméletének igazolására szolgáló mérésekben.) A távolba hatást végül is a közbeeső térnek kell közvetítenie, amely tehát még a világűrben sem lehet „üres".

Az egész bevezető tanulsága annyi, hogy egyrészt: bár az elektrosztatikára már van a fizikusoknak pontos magyarázatuk, a fizikának még mindig „hija van", másrészt: olykor igen sok mélyebb ismeretről lemondunk, amikor „fenomenologikus tárgyalásban" tanítunk meg egy tárgyat.

Ebben a hónapban, ismét a Tudományszervezési és Informatikai Intézet választékából — a sorozat múlt havi számában leírtaknak megfelelően „második generációs"-nak nevezhető —, elektrosztatika témájú oktatóprogramot elemzek. A programcsomag tehát átírás HT iskola-számítógépről, de a szerző ebben is figyelembe vette az új gép adta lehetőségeket, amennyiben egy külön grafikás programot is hozzáfűzött a programcsomaghoz. A programcsomag természetesen az elektrosztatika fenomenologikus leírásának támogatására készült. Nem is lehet másképpen. Ez a mai középiskolás szinvonal, és nem csak nálunk, Magyarországon. Bele kell nyugodni, hogy a tudomány rohamléptű fejlődésével csak a hozzá nem értők dilettantizmusa fejlődik egyenes arányban.

A programleírás elég vékonyka, hozzá az erővonalak egylapnyi szórólapos információját kaptam meg, de a végleges dokumentáció készítése folyamatban van. A leírás mellé gondolhatók még persze az iskolai fizikakönyvek, amelyekkel együtt meg lehet érteni a programok működését és célját.

A programcsomag elég kis részt fed le még az elektrosztatikából is, amit valószínűleg az a tudat befolyásolt, hogy a programoknak esetleg C16-on is futniuk kellene. Ezt nem próbáltam ki, de néhány program elég rövid algoritmusúnak tűnik.

Az Elektrosztatika programsorozat kevésbé, az Erővonalak program alaposan kihasználja a Plus/4 grafikai tulajdonságait. Az utóbbi igen szemléletesen tudja bemutatni a ponttöltések közötti erőteret. Meg lehet választani az egyes töltések helyét a képernyőn, és azokra több-kevesebb elektromos töltést rakhatunk. A kialakuló erővonalak megrajzolása elég hosszadalmas, különösen nagyobb számú töltés felrakása esetén. (Én nem próbáltam ki csak négy töltéssel, de ez negyedórába került. Órai bemutatásnál tehát nemigen lehet túl bonyolult eseteket választani! Szakköri foglalkozáson és egyéni tanulásnál már jobb a helyzet.)

A program „régi" része egy menüvezérelt rendszer, amely nyolc alprogramra támaszkodik. A sorozatban találkozhatunk az elektrosztatika történetével, az elektroszkóppal bemutatott elektromos megosztás jelenségével, a Coulomb-törvénnyel, a kuriózumnak számító Millikan-kísérlettel (amely a cikk szerzőjének rossz álmaiban szokott előjönni, mert annak idején az egyetemen szemtől szembe találkozott a kísérlettel! — számítógéppel persze könnyebb imitálni), további érdekességként szerepel Franklin villámhárítója, majd statikai feladatok következnek. Van egy program a potenciálgörbékről, és a sorozatot újabb kuriózumként a Van de Graaf-féle generátor zárja (de szerintem az eredetivel játszadozni izgalmasabb — úgy emlékszem, még a mi vidéki gimnáziumunkba is jutott egy „zseb-Van de Graaf"). A programok többnyire jól működnek, de néhány helyen nem védekeznek eléggé a helytelen adatok ellen.

ZSADÁNYI PÁL

ÖSSZEFOGLALÓ ADATOK

Forgalmazó:	**Tudományszervezési és Informatikai Intézet**
Terméknév:	**Elektrosztatika és Erővonalak**
Szerző:	**Tolvaj László**
Géptípus:	**Plus/4**
Hordozó:	**kazetta vagy lemez**
Dokumentáció:	**rövid, a célnak megfelelő**
Ár:	**650 Ft (kazettán) (floppy esetén + floppyköltség)**

MINŐSÍTŐ ADATOK

Kezelhetőség:	**kiváló**
Teljesség:	**közepes**
Dokumentálhatóság:	**közepes (jelenleg)**
Használhatóság:	**jó**
Ár/teljesítmény:	**jó**
Összbenyomás:	**jó**

**Donát János:
WordStar
(Budapest, 1988.
Műszaki Könyvkiadó—
Novotrade,
49 oldal. Ára: 120,— Ft.)**

A Lapozgató sorozat új kötete a WordStart, *az egyik legnépszerűbb szövegszerkesztő programot mutatja be olyan felhasználók számára, aki IBM PC vagy azzal kompatibilis számítógéppel és MS-DOS operációs rendszerrel dolgoznak. A program más gépeken és más operációs rendszerek (például CP/M) alatt is fut. Igen könnyen előállíthatók és kezelhetők vele mindazok a szövegszerű állományok, amelyek végső formája egy nyomtatott alakban megjelenő dokumentum.*

**Baumgartner, R.—Hansjakob, S. —Praxl, W.:
Turbo-Pascal. Elmélet
és gyakorlat
(Budapest, 1988.
Novotrade, 271 oldal.
Ára: 290,— Ft.)**

A kötet — immár második kiadásban — azokat az ismereteket tartalmazza, amelyekre a Turbo-Pascallal kapcsolatban, a rendszerközeli programozásban a szerzők szert tettek. E rendkívüli fordítóprogram a felhasználónak új lehetőségeket mutat be. Sok olyan probléma, amelyhez korábban assembler programozásra volt szükség, most magas szintű programnyelv segítségével áttekinthetően megoldható.

Mivel az ilyen típusú programozáshoz részletesen ismerni kell az adott operációs rendszert és az alkalmazott programnyelvet, a könyv első részében az ehhez szükséges alapok találhatók. A második rész a gyakorlat, programozási példákkal, amely a programozás alkalmazott fogásait és módszerét mutatja be.

**Áts László:
Superbase 64 kezdőknek
és haladóknak
(Budapest, 1988.
Novotrade, 535 oldal.
Ára: 280,— Ft.)**

A Superbase 64 adatbázisrendszert a Precision Software Inc. fejlesztette ki a Commo-

ADOK—VESZEK —CSERÉ

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közleteknek gépelt soronként (60 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,-Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

Commodore Plus/4 datasette-tel, 150 programmal (15 500 Ft) és 1541-II lemezegység 10 originál 3M diszkkel (21 000 Ft) eladó. Együtt olcsóbb! Bedegi, Kaposvár, Németh I. fasor 34. fsz. 3. 7400

C64-re színvonalas programok (játék, demo, grafikus) eladók. Csere is lehetséges. Keresem a következő programokat: Desk Pack 1, Writer's Workshop, Geospell, Geofont, Fontpack 1, Geopublish. Sütő Zoltán, Hódmezővásárhely, Kohán Gy. u. 3. 6800

Teljesen új Commodore 64-esemet és 1541-es floppymat (mindkettő új típusú) plusz szükség szerint cartridge-t (reset, fast load, monitor) és lemezeket programokkal eladom. Sürgősen pénzre van szükségem. A leveleket árajánlattal (részletre is!) a következő címre kérem: 2661 Balassagyarmat, Pf. 118

C16 programok eladók (12 Ft/db) magnókazettára. Klemm Pál, Zebegény, Dózsa György út 28. 2627

C64-es programok olcsón eladók! Nagy Dániel, Szászhalombatta, Pannónia út 7. 2440

Gyári 64 k Homelab-4, debuggerrel, assemblerrel, duplapontos aritmetikával, programokkal eladó. Árajánlatokat kérek: Molnár Gyula, Oroszlány, Népek barátsága út 61. III/3. 2840

PLASTIC programcsomag Enterprise-ra kapható (adatkezelés, szótár, táblázatkezelés, diagramkészítés, ékezetes és cirill betűk). Varga Zoltán, Székesfehérvár, Lövölde u. 9/C. IV/2. 8000

Commodore 64-es RESET kapcsolóval, magnóval, több száz programmal (10 kazettán) és szakkönyvekkel eladó. Irányár: 28 000 Ft. Heidrich Attila, Leninváros, Bartók Béla u. 4. V/4. 3580

Magyar nyelvű dokumentáció eladó az alábbi programokhoz: Printfox, Charakterfox, Geos, Platine-64 stb. Kérésre listát küldök. Honti József, Csákvár, Május 1. út 11. 8083

Spectrum programkazetták (48/128 k-s) a legújabb angol slágerlistákról eladók. Tájékoztatót válaszborítékban küldök. Horváth Péter, Siklós, Pf.: 129. 7800

Spectrum játék- és felhasználói programok eladók. Mindig a legújabb szoftverek! Katalógusért küldjön válaszborítékot! Frim András, Pécs, Zrínyi u. 1. 7621

Spectrum (48, 80 k) disassembler eladó 350 Ft (10 parancs, 80 k teljes kezelése). Kérjen tájékoztatót! Interfész I. 4500 Ft-ért eladó. Polgár József, Kecskemét, Balaton u. 20. fsz. 30. 6000

ZX-Spectrum (48 k) alig használt gépemet programozási kézikönyvvel eladom. Irányár: 15 000 Ft. Takács Lajos, Kisvárda, Mártírok útja 18. II/3. 4600

Utánvéttel eladó a '64-er magazin 1986-os, 1987-es évfolyama, 1988-as számai, az Amiga magazin, Happy Computer példányok 99 Ft/db, PC magazin 40 Ft/db. Részletes listát küldök! Kőházi Zoltán, Székesfehérvár, Mikszáth Kálmán u. 5. II/55. 8000

VESZEK

Bármilyen üzemképtelen vagy használt számítógépeket, floppykat, nyomtatókat, részegységeket stb. veszek. Kaszala Sándor, Kunszentmiklós, Tavasz út 7. 6090

Az Enterprise technikai ismertető című könyvet megvenném. Kecskés Norbert, Kazincbarcika, Vajda J. u. 16. I/1. 3700

CSERÉLEK

C64-es programokat cserélek kazettán. A válaszokat listával kérem. Bárd Attila, Budapest, Népfürdő u. 21/F. 1138

C64-es gépre cserélek játék- és felhasználói programokat. Csak lemezen. Listát kérek! Már András, Budapest, Derék u. 21. 1016

C16, Plus/4 és C116 játék- és felhasználói programokat cserélnék kazettán. Listát kérek, cserekazettát küldök. Szabó Tibor, Miskolc, Éder György u. 12. 3527

C64 + floppy + kazettás egység + monochrom monitor, 106 lemez teli új játékprogramokkal, sok kiegészítő tartozékkal, igen olcsón eladó. Zörgő Zsolt, Miskolc, Bokány D. u. 36. I/2. 3529. Telefon 18 óra után: 06-46-64909

C16 bővítővel, magnóval, sok programmal, dokumentációval, átalakítóval 11 ezer forintért eladó. Varró Péter, Vésztő, Kossuth u. 53. 5530

C16 eladó + magnó, joystick, játékprogramok. Ár megegyezés szerint. Telefon: 471-011

Commodore 64-hez GYORS-HÁTTÉRÁR cartridge. Kapacitása 2-31 kbájt. A GYORS-HÁTTÉRTÁR-ba maximum 7 db, célszerűen gyakran használt program vihető be. A gép bekapcsolása után menüvel jelentkezik be, és gombnyomásra a kiválasztott program azonnal fut. Javasolt programcsomagok: Turbo tape, Assembler, Monitor, Supergrafik, Help plus, Turbo másoló, File másoló 1699 Ft és Turbo tape 699 Ft.Hozott programok elhelyezése a GYORS-HÁTTÉRTÁR-ban. Trompler László, Budapest, Attila u. 22. 1201. Telefon: 287-493 este

C64 + VC1541 + SP100 + Dataset + teljes irodalom + 2500 felhasználói program eladó. Érdeklődni: 378-363 18-22 óráig.

C64-tulajdonosok! Saját fejlesztésű COPY! Egy másoláshoz elég egy gombnyomás, szemben a szokott 5-6-tal. 30 karakteres név. Egyéb szolgáltatások. Ára: küldött kazettára 100 Ft utánvéttel. Böke László, Székesfehérvár, Münnich ltp. 22. 3/3. 8000

C64 + VC1541 + joystick + programok lemezen eladók. Árajánlatokat levélben várom. Irimi János, Hajdúnánás, Ady Endre krt. 33. I/6. 4080

Commodore Plus/4 magnóval, joystickkal, valamint egy Datasette magnó eladó. Tel.: 347-989

Enterprise számítógép magnóval, 2 joystick átalakítóval és 10 db programmal eladó, decemberig garanciával. Árajánlatokat kérek. Csabai Károly, Kocsér, Kossuth Lajos u. 22. 2755

Enterprise programokat kitűnő minőségben adok-veszek és cserélek. Benkő Sándor, Nyíregyháza, Rákóczi út 23. I/5. 4400

dore 64 típusú személyi számítógépre. Segítségével sokféle nyilvántartási, ügyviteli adatfeldolgozási feladat oldható meg gyorsan és hatékonyan.

A Superbase 64 maximális kényelmet nyújt a laikus felhasználónak, ugyanakkor mint fejlesztőrendszer, nagy, generatív erejű programnyelvet kínál a gyakorlottabb programozóknak, mivel a rendszerhez két hozzáértési szinten lehet hozzáférni.

A könyv első fejezetei elsősorban azoknak az olvasóknak szólnak, akiknek szükségük van a számítástechnikai eszközök használatára, de félelem, idegenkedés riasztja el őket a munkavégzésüket megkönnyítő „okos" gépektől. Márpedig a számítástechnika mindenkiért — értük is van. Ezért a szerző nem a programozás, nem a matematika, még csak nem is az elektronikus áramkörök felől közelít a számítógéphez, hanem egy általános adminisztrátor, humán beállítottságú szakember, sőt egy háziasszony mindennapjaiban előforduló problémák irányából. Erre a Superbase 64 adatbázisrendszer kiváló lehetőséget nyújt.

A kötet egyes fejezetei a munkához szükséges eszközöket, a számítógépet és tartozékait mutatják be, példaanyagon keresztül vezetik be az olvasót a Superbase-menük felhasználásával megvalósítható, igényesebb alkalmazásokba. Önálló fejezet foglalkozik a különféle alkalmazásokkal. A példák a háztartás, a tudományos kutatás és az ügyvitel területéről származnak.

A könyv elsősorban azoknak szól, akik mindennapi munkájuk során a Superbase 64 segítségével még hatékonyabban akarják a Commodore 64 számítógépüket használni.

Nestle, F.—Ostertag, E.: BASIC, LOGO, PASCAL (Budapest, 1988. Novotrade, 149 oldal. Ára: 149,— Ft.)

A közölt feladatokat mindhárom nyelven kétféle programmal oldja meg: alapprogrammal és felhasználói programmal. Alapprogramon a legegyszerűbb algoritmusra redukált feladatot érti. Ezek a programok a működésük ismerete nélkül nem használhatók. A felhasználói program a futás során megadja a felhasználó számára mindazokat az ismereteket, amelyek a program használatához szükségesek.

A könyv elsősorban a programnyelvtől független, tipikus problémamegoldásokat akarja bemutatni. Ez azt jelenti, hogy a felhasználó szempontjait figyelembe véve oldja meg a feladatot, mintegy a képernyővel „beszélgetve".

Aki még soha nem dolgozott személyi számítógéppel, a példákból megtudhatja, hogyan fordítható le a feladat egy másik programnyelvre. Eközben megismerkedhet a nyelv szerkezetével is. A könyvben szereplő programok — néhány kivételtől eltekintve — bármely számítógépen futtathatók, mindhárom nyelven.

Mesterséges intelligencia a gyógyszerkutatásban

A hazai CompuDrug Kisszövetkezet és az amerikai Kiser Research Incorporation cég 300 ezer dolláros alaptőkével vegyesvállalatot alapított Texasban, a kisszövetkezet legújabb szoftverének forgalmazására, amely a kémia szakterületén — főként a gyógyszertervezésnél, -fejlesztésnél és -gyártásnál — a mesterséges intelligenciára alapozott megoldások segítségével információkat ad a szakembereknek. Alkalmazásával a gyógyszerkutatók már a vegyület tervezésénél megtudhatják, miként alakul át az új anyag az élő szervezetben, a betegekben. A kisszövetkezet — amely elsősorban kémiai programok fejlesztésével foglalkozik és ezeket mintegy 18 országban, köztük Indiában, Japánban, az Egyesült Államokban és Kanadában forgalmazza — arra számít, hogy e legújabb szellemi terméke is sikert arat, mivel eddig meg nem oldott problémára ad választ. A CompuDrug hosszabb ideje együttműködik a floridai egyetemmel a kutatásban és fejlesztésben. Ennek eredményeként a tervek szerint még az idén újabb vegyesvállalatot alapít, és szó van arról, hogy hasonló vállalkozást hoz létre Kínában is.

STRUKTURÁLT FORTRAN

A RATFOR, a FORTRAN 1988-ban megjelent kiterjesztése lehetővé teszi, hogy a kissé nehézkes, de egyébként jól kiegészített és viszonylag hatékony FORTRAN nyelvet a korszerű követelményeknek megfelelően használjuk. Kihasználhatók a strukturált programozás előnyei, könnyebb a programtervezés és -kódolás, messzemenően elősegített az olvashatóság és az áttekinthetőség, a korábbinál kevesebb próbálkozással valósítható meg a szoftverfejlesztés. Az SZKI által piacra hozott RATFOR bármelyik FORTRAN-változathoz használható, így egyetlen verzióval sincs összeépítve. Mint fordítóprogram abban tér el a szokásostól, hogy nem gépi nyelvre, hanem FORTRAN-ra fordít. A keletkező FORTRAN programból a szokásos eszközökkel (FORTRAN szerkesztő, fordító) kell a futtatható programot létrehozni.

C64-es programakat cserélek kazettán. Listát kérek. Belegrai Tibor, Bodajk, Petőfi u. 130. 8053

Enterprise programakat cserélek. Listát adok és kérek. Sala István, Budapest, Hársfa sétány 17. 1203

C128, CP/M, C64 felhasználói és játékprogramokat, leírásokat cserélek. Mészáros Károly, Pécs, Semmelweis u. 36. 7623

Primo programakat cserélek. Keresem azok ismeretségét, akik hardverbővítést végeztek gépükön. Varsányi Gábor, Nagyatád, Aradi u. IX/C. IV/3. 7500

C64-es programakat cserélek kazettán és lemezen. Listát kérek és küldök. Borsodi Zoltán, Budapest, Kárász u. 4. II/1. 1131

Enterprise programakat cserélek, elsősorban lemezen (VC 1541). Veresegyház, Pf. 28. 2112

ZX-Spectrumra (48 k) színvonalas játék és felhasználói programokat cserélek. Többek között keresem a Navy Moves és Trivial Pursuit programokat. Kajtár Zsolt, Bonyhád, Zalka Máté u. 19. 7150

Atari 800XL játékprogramokat cserélek kazettán, listát adok és kérek. Nagy Károly, Budapest, Havanna u. 73. 4/16. 1181

C16, Plus/4 programokat cserélek. Kovács Árthur, Szeged, Engels u. 11. 6771

C64-es játékprogramokat cserélek kazettán. Listát kérek! Takács Róbert, Budapest, Szuglói körvasútsor 7. 1144

C64-es programokat cserélek lemezen és kazettán. Komjáthy Attila, Siklós, Felszabadulás u. 96. 7800

Commodore-ra, Enterprise-ra, Spectrumra évente kb. 1,2 Mb-nyi programot cserélek rendszeresen (lemezen, kazettán). Póka László, Budapest, Sáfrány u. 50. VII/80. 1116

Enterprise programokat cserélek kazettán. Listát kérek! Három István, Szolnok, Barátság út 14. 5000

MSX rendszerű számítógéphez programokat cserélek. Jancsurák István, Miskolc, Dráva u. 7. 3528. Telefon: 06/46/ 12634.

Ez a rovatunk **KODEX 2000** szövegszerkesztővel készült.

A segítség szerkesztője

A Help Master nevű új hazai program végigkísérheti egy szoftver életét a születéstől a fejlesztésén át, egészen a felhasználó általi alkalmazásig. A program egy helpállományt szerkeszt és kezel. A szoftverfejlesztőknek jól jön, ha a program készítésekor olyan mankójuk van, amely csökkenti az ilyenkor egyébként is fennálló bizonytalanságot. Menet közben ez a segédlet folyamatosan fejleszthető újabb információkkal a különféle funkciókról, nevekről, parancsokról. A programtermékkel így együtt nő föl, s elkészültekor azonnal készen van a help is, amely már menet közben olyan felépítésűvé alakul, hogy valóban válaszoljon a felmerülő kérdésekre. A fejlesztés lehetővé teszi azt is, hogy a különböző felkészültségű felhasználók más-más help képernyővel dolgozzanak. A Softinvest által piacra hozott, az IBM PC-vel kompatibilis gépeken futó, memóriarezidens Help Master 250 kbájt operatív tárat foglal el.

Pontvadászat

Rejtvénysorozatunk minden fordulójában két feladatot közlünk: az első logikai, matematikai tudást, a második számítástechnikai alapismereteket is igényel.

A feladatok után közöljük az elérhető maximális pontszámot. A részmegoldásokat is pontozzuk.

A pontgyűjtést, vagyis a pontvadászatot az esztendő végén zárjuk. A legjobb tíz versenyző nevét magazinunkban közzé is tesszük, ők lesznek azok, akik könyvutalványt is kapnak.

A helyes megoldások a feladatok közreadása után két lapszámmal később jelennek meg, így a pontvadászoknak jut idejük a gondolkodásra.

Beküldési határidő: 1988. december 16.
Címünk: Mikroszámítógép Magazin Szerkesztősége
1371 Budapest, Pf. 433.

Az 1988/5. szám 1. feladatára hat, 2. feladatára nyolc megfejtés érkezett be. Ezek közül az 1. feladat megfejtési százaléka, a maximális 6 pontot véve 100 százaléknak, 90 százalék, a 2. feladaté, a maximális 4 pontot véve 100 százaléknak, 100 százalék.

Jó vadászatot kíván a feladatok összeállítója:

dr. Hoffmann Tibor

1. feladat

Már a XVII. század óta foglalkoznak az ún. feltöltési-szétosztási probléma sok változatával. Ezek közül most egy egyszerűt vetünk fel.

a) Legyen három edényünk. Az egyik *4* literes, a másik *3* literes és a harmadik *2* literes. A három edényben összesen *6* liter folyadék van (például úgy, hogy az első és a harmadik edény tele van, a második üres). A folyadék áttöltögetésével osszuk el a három edénybe a *6* litert úgy, hogy edényenként *2-2* liter folyadék legyen, de az eljáráshoz nincs más mérőeszközünk, mint annak megállapítása, hogy valamely edény tele van-e vagy üres. Ha az egyik edényből a másikba öntjük át a folyadékot, a feltöltendő edény teleségét a töltés úgy jelzi, hogy azonnal leáll az áttöltési folyamat, és így nem vész el folyadék.

Adjunk egy lehetőleg *rövid áttöltési lépéssorozatot*, amellyel ez elérhető. *Hány áttöltést* igényel ez? (2 pont)

b) Az előbbi hagyományos feladatot módosítsuk most úgy, hogy a *4* literes edényben lévő A fajta folyadék különbözik a *2* literes edényben lévő ugyanolyan sűrűségű B fajta folyadéktól, de az A és B folyadék tetszőleges arányban keverhető. Ezzel a kiindulási feltétellel ugyancsak osszuk el a keveréket három egyenlő mennyiségű részre, de további követelményként kívánjuk meg azt is, hogy az A és B *keverési aránya is ugyanaz legyen* a három edényben. Tételezzük fel, hogy egy bizonyos tartalmú edényben a folyadékok azonnal összekeverednek. (4 pont)

2. feladat

Egy számítógépen a következő műveleti idők vannak:

a) Az operatív tár egy adott indexű tömbelemének a veremtár (stack register) egy adott indexű tömbelemébe való átvitele *15* μs.

b) Az operatív tár tömbindexértékének *1*-gyel való növelése *2* μs.

c) A veremtár tömbindexértékének *1*-gyel való növelése *1* μs.

d) A veremtár egyik tömbelemének a veremtár egy másik tömbelemével való megszorzása az előbbi helyén *20* μs.

e) A veremtár egyik tömbelemének a veremtár egy másik tömbeleméhez való hozzáadása az utóbbi helyén *30* μs.

f) Az operatív tár tömbindexének értékadása *2* μs.

g) A veremtár tömbindexének értékadása *1* μs.

A veremtár mérete *69* szó. Ebben *4* kisméretű szó (a tömbindexek helye), egy nagyméretű szó (a végeredmény tárolására) és *64* szó van a tömbelemek tárolására.

A számítógépen az esetleges ciklusszervezési időket elhanyagolhatjuk.

Számítsuk ki annak az időszükségletét, hogy az operatív tárban levő két, egyenként *32* elemes tömb *skalárszorzatát* a veremtár valamely elemében hozzuk létre. (6 pont)

Mennyire csökken ez az idő, ha a hardver olyan, hogy a veremtárban levő egyik *32* elemes tömbnek a másik *32* elemes tömbbel való elemenkénti szorzását a gép egyszerre végzi el (*32* párhuzamosan működő processzorral) és helyezi el az egyik tömb elemeinek a helyén? Ennek a műveletnek az ideje *20* μs. (4 pont)

Az 1988/9. szám feladatainak megoldása

1. feladat

A legnagyobb k-jegyű szám nyilván

$$(p-1)(1+p+p^2+\ldots+p^{k-1})=p^k-1.$$

A kívánt összeg tehát

$$\sum_{k=1}^{n}(p^k-1)=\frac{p(p^n-1)}{p-1}-n. \qquad \textit{(4 pont)}$$

2. feladat

A p alapú számrendszerben egy M számot $\frac{\ln M}{\ln p}$ számjeggyel, vagyis az M és a p természetes logaritmusainak hányadosával ábrázolhatunk. E számjegyek mindegyike p állapotú lehet, és így az összes kijelzési állapotok száma

$$\frac{p\ln M}{\ln p}.$$

Ezek szerint p függvényének minimuma van, ha

$$\frac{1}{\ln p}-\frac{1}{(\ln p)^2}=0,$$

aminek megoldása $\ln p=1$,

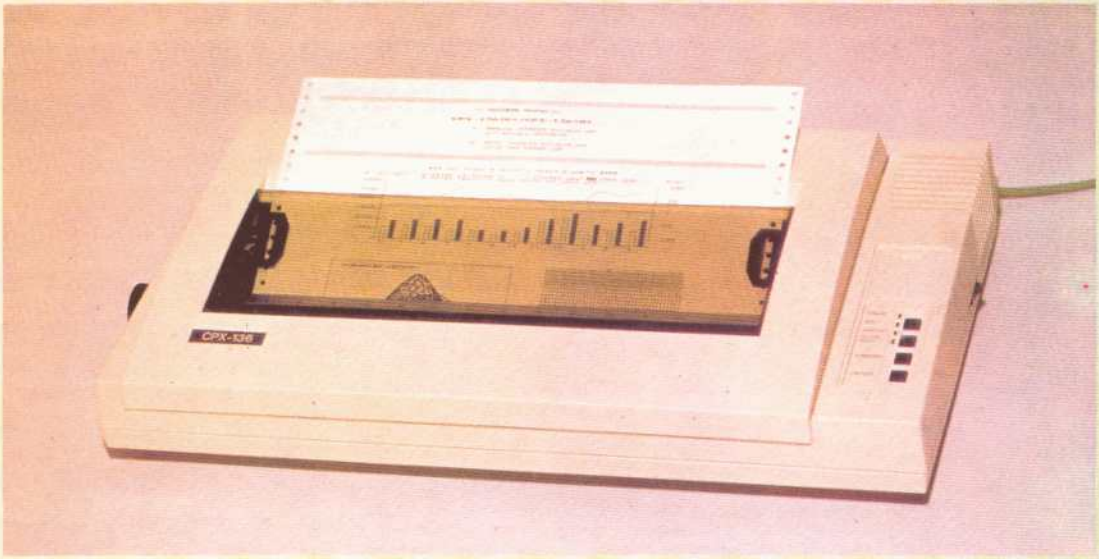
vagyis $p=e=2{,}7183\ldots$

a természetes logaritmus alapszáma. Minthogy reális építéshez csak egész számú p-vel dolgozhatunk, $p=3$ vagy $p=2$. $p=3$-ra $\ln M$ faktora

$$\frac{3}{\ln 3}=2{,}7307\ldots,$$ míg $p=2$-re

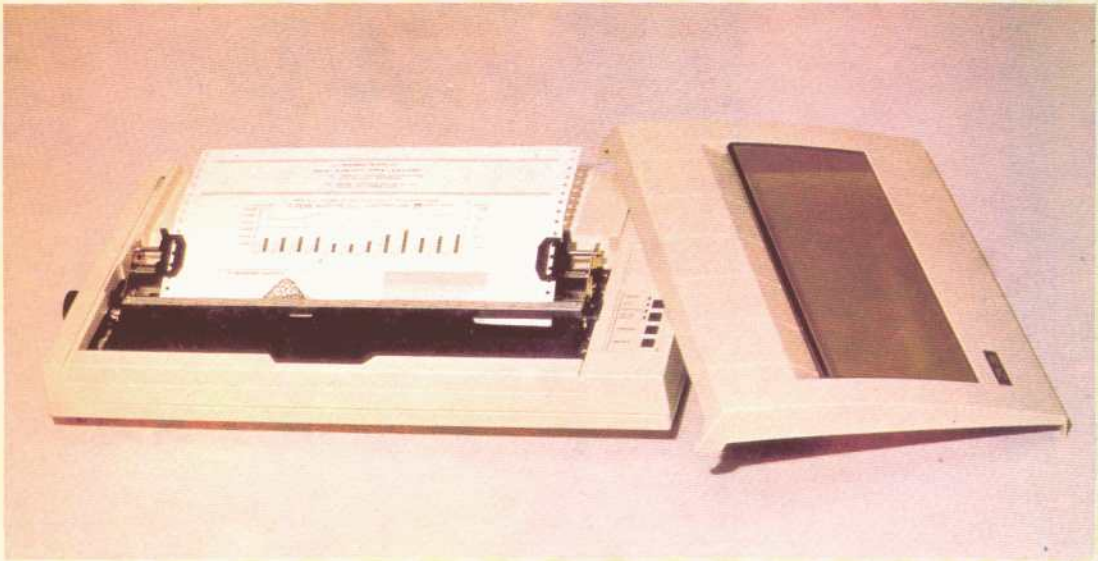
$$\frac{2}{\ln 2}=2{,}8854\ldots$$

Tehát a $p=3$ lenne a kedvezőbb, s valóban ilyen (háromállapotú elemmel működő) gépet konstruáltak ugyan, de a gyakorlatban használt gépek mégis a $p=2$ bináris változatra épültek, főleg a félvezető elemek biztonságosabb kétállapotú működése miatt. (8 pont)



M6000

A Műszertechnika Kisszövetkezet — a nagy sorozatú IBM PC-vel kompatibilis mikrogép gyártásán kívül — az idén megkezdte az ezekhez a gépekhez teljesítményben is illeszkedő mátrixnyomtatók sorozatgyártását. Az M6000 típusjelzésű nyomtatónak hat nyomtatófeje van, kilencfajta betűtípus használatát teszi lehetővé mind a hagyományos, mind a közel levélminőségű kiíráskor. A felhasználó által definiálható 116 karakter nagyfokú rugalmasságot tesz lehetővé. Az M6000 kompatibilis az FX 1000-rel, de annál mintegy háromszor gyorsabb: a 214 sor/perc teljesítménye a párhuzamosan dolgozó hat nyomtatófejnek köszönhető.



ROBOTRON „ERIKA 3004 ELECTRONIC" ELEKTRONIKUS KIS ÍRÓGÉP

INFORMÁCIÓ:

Író- és Számológép Osztály
Bp.IX.Dimitrov tér 14.
T: 175-163, 175-081
175-294

MIGÉRT

Az „Erika 3004 electronic" kisméretű, formatervezett, a mai igényeknek megfelelő, hordozható, irodai elektronikus írógép.
Egybeépített táskájával és kiemelhető fogantyújával könnyen és biztonságosan szállítható, és így az üzleti életben, az irodában és otthon gyors, megbízható igénybevételt tesz lehetővé.
Egyszerűen cserélhető margaréta kerekei egyéni írás megválasztását teszik lehetővé, így minden egyes levél a feladó jellegzetes „névjegyévé" válhat.

Ára: 20 100 Ft + ÁFA

MIGÉRT

MŰSZER- ÉS IRODAGÉP-ÉRTÉKESÍTŐ VÁLLALAT

KÍNÁLATA:

MIGÉRT

SANYO MN—1000

minikazettás diktafon

— Kis méretben nagy teljesítmény
— Elektronikus jegyzetfüzet
— Hű gondolatrögzítő, tökéletes üzenethagyó

Irányár: 6 200 Ft + ÁFA — raktárról szállítható

Felvilágosítás: MIGÉRT Író- és Számológép Osztály
Bp. IX., Dimitrov tér 14.
Tel.: 175-163, 175-081, 175-294